Volume Thirty-Six Number One!

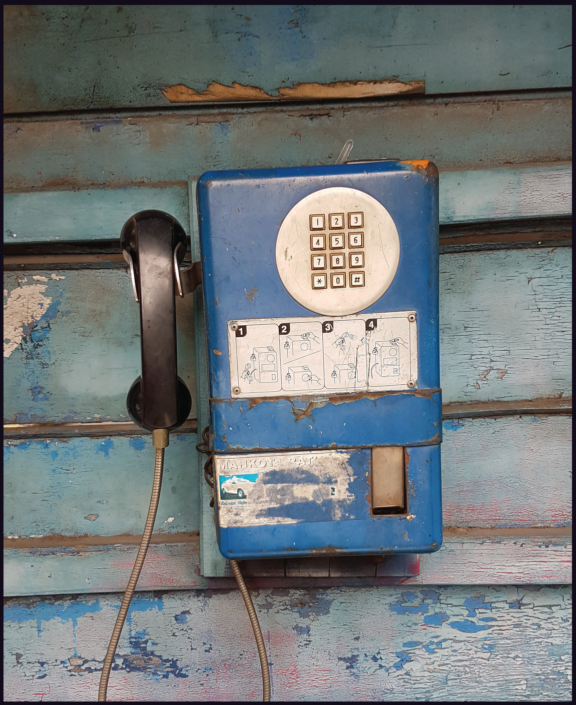
DIGITAL EDITION Spring 2019

2600

The Hacker Quarterly

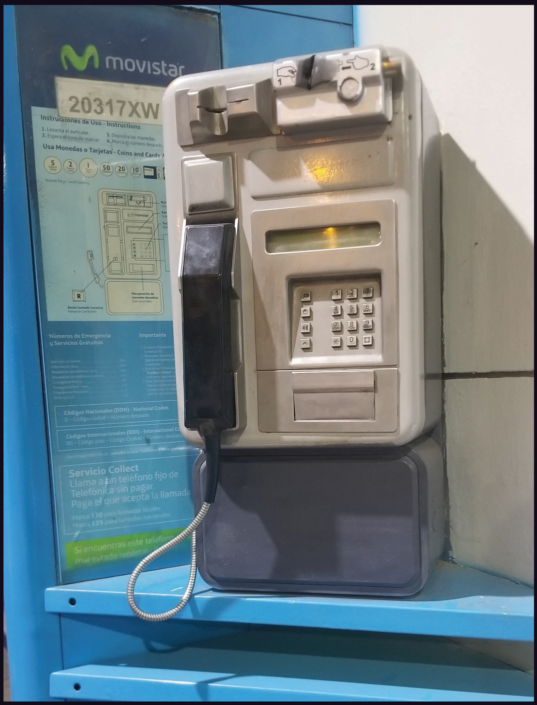
Ceci n'est pas un ordinateur

Blue Payphones



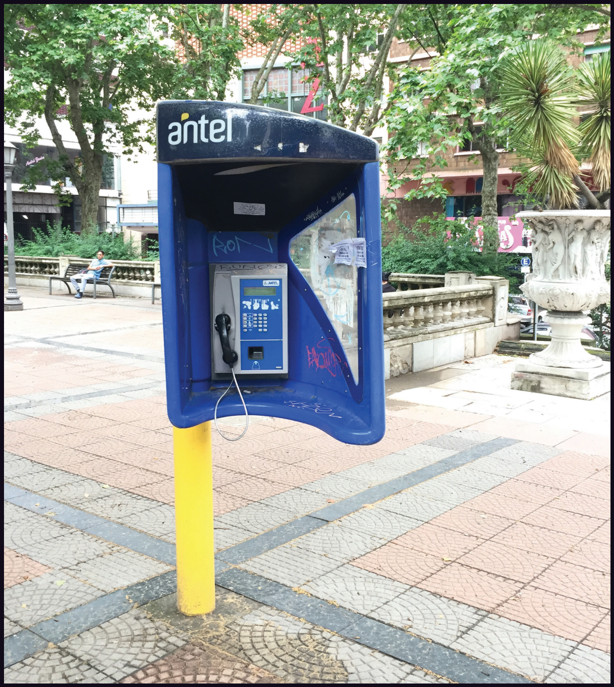
Indonesia. Seen in the city of Solo, this old blue box is sadly no longer in operation.

Photo by Carl Rudnert



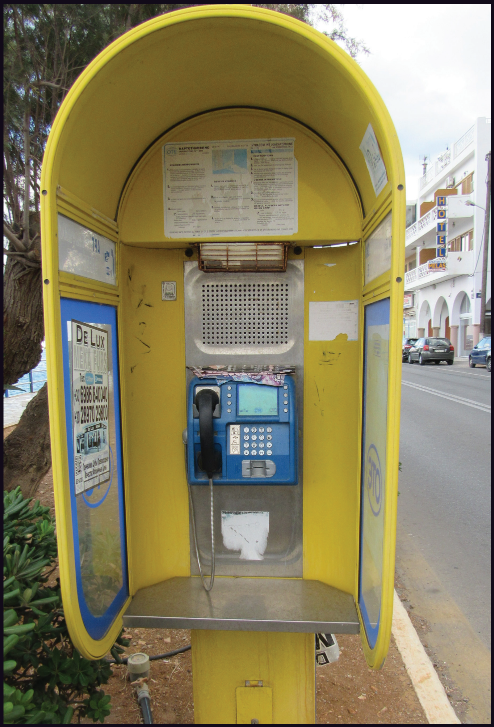
Peru. A very common model, this one was found in Chiclayo. While it's not blue itself, it has enough of that color surrounding it to qualify.

Photo by Elias Mirror



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



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

Got foreign payphone photos for us? Email them to **payphones@2600.com**.
Use the highest quality settings on your digital camera! (Do not send us links as photos must be previously unpublished.) (More photos on inside back cover)

Errata

MMDC

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.

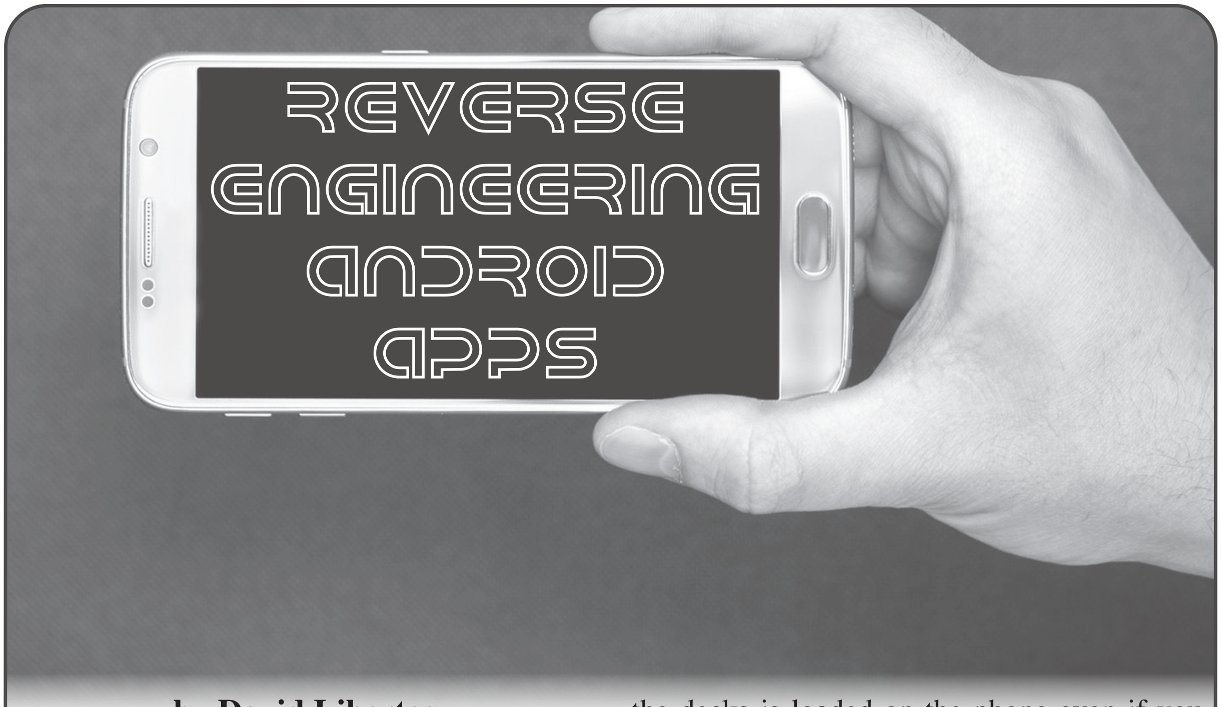
The same can be said for variety *within* the tech we use, regardless of age. Having a wide assortment of devices, operating systems, and telecommunication options ensures that we will develop and evolve by recognizing what's good in one and finding a way to apply it in another. When we become zealots 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.

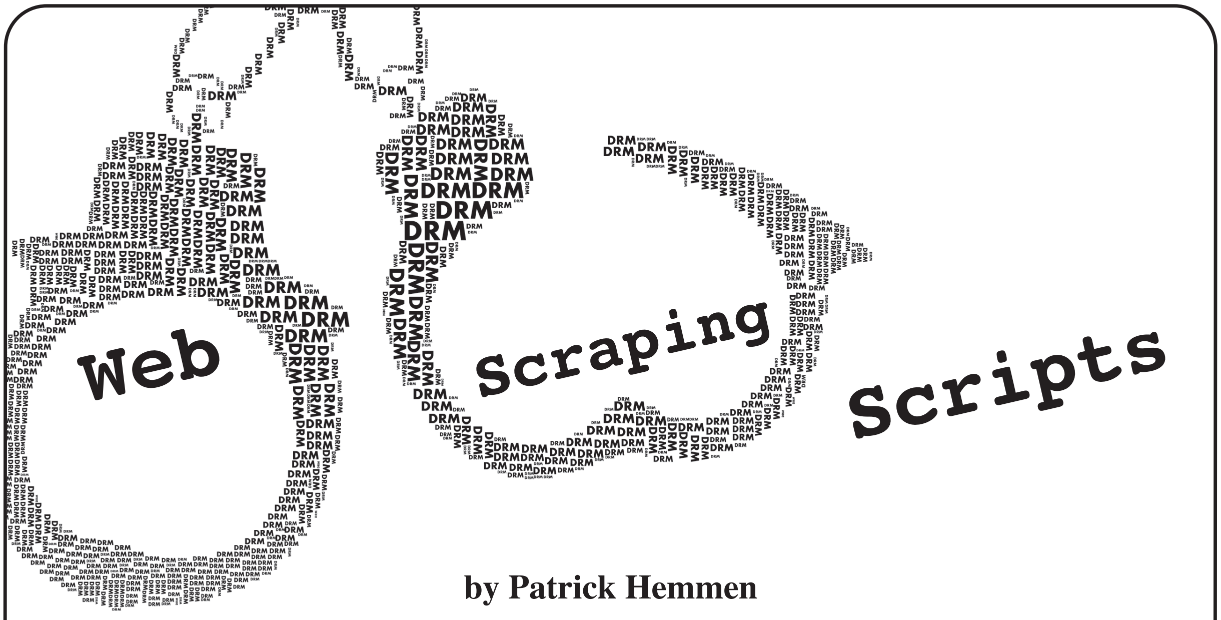
This one was a repeat visit. There had been a recurring hawk problem. Hawks build giant nests on cell towers, and it's illegal to disturb them. Six months ago, a crew brought in new equipment, but a hawk took up residence before it could be installed. Wildlife specialists monitor the sites until the 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
```

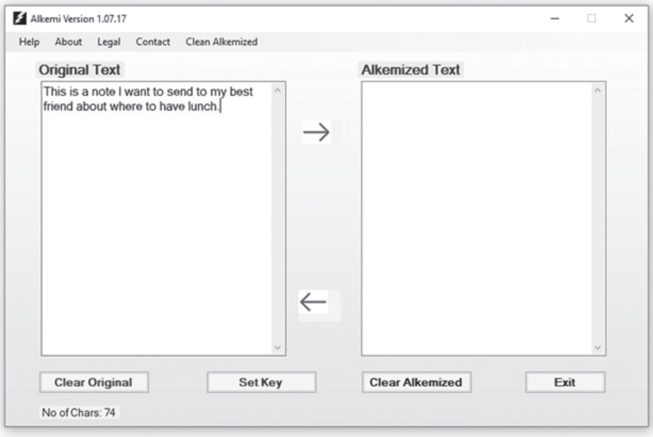
A GUIDE TO ALKEMI

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.

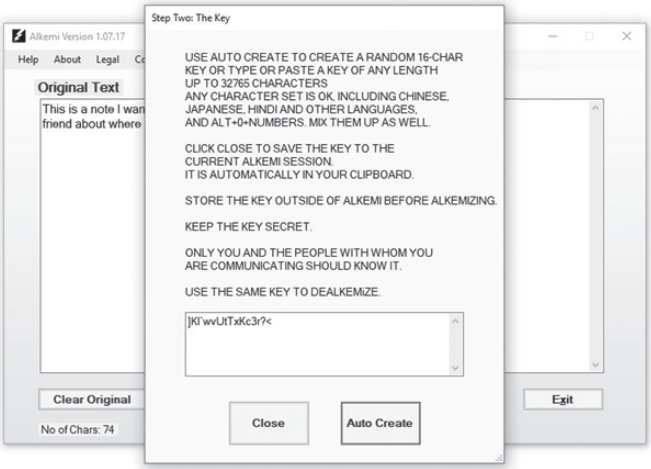
Here's how it works: Type or paste text into the Original Text box up to 12,000 characters. Any language that uses characters is OK, including Chinese, Japanese, Hindi, Hebrew, Arabic, English, etc.



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.

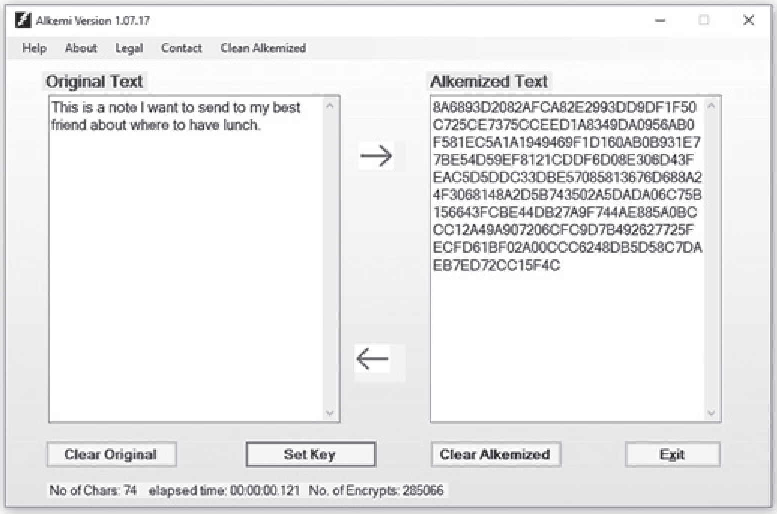
KEEP THE KEY SECRET.

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

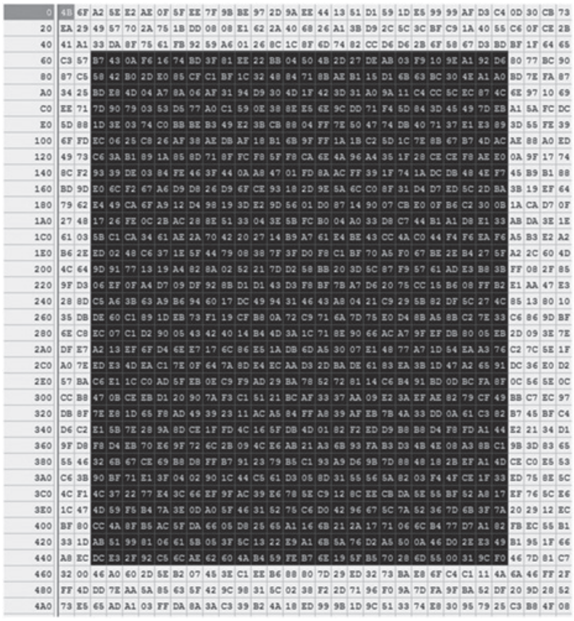
Being just text, it can be sent anywhere text can be sent, like Twitter, email, etc. You can tweet your followers with the Alkemized text and, since only they have the key, only they can decrypt it. Text data, of course, is just data like anything else.

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);

 }
```

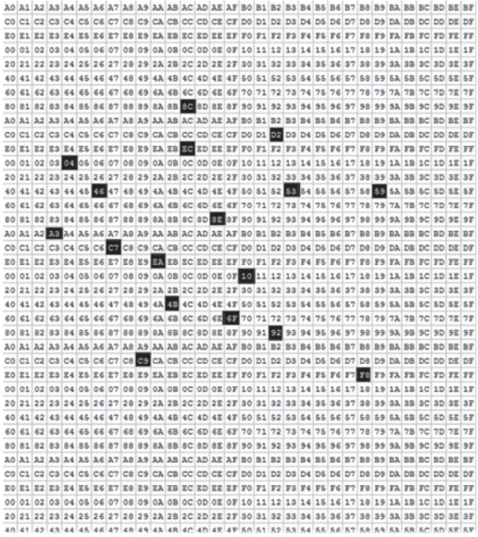
Some procedures create a byte array based on the values from the key, others from a "matrix" like extraction:



```
   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:

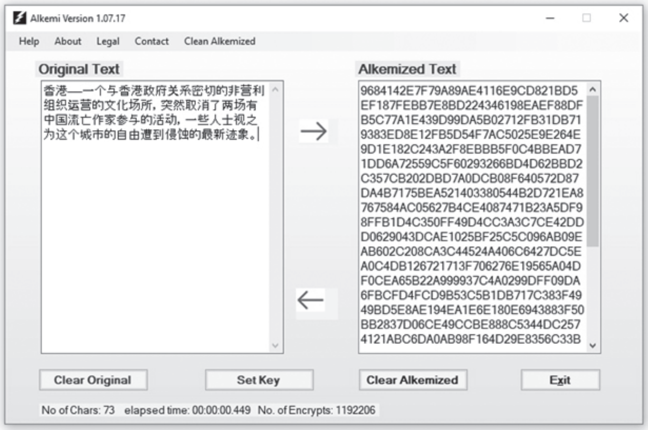


A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
00	01	02	03	04	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:



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。

```
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.

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.

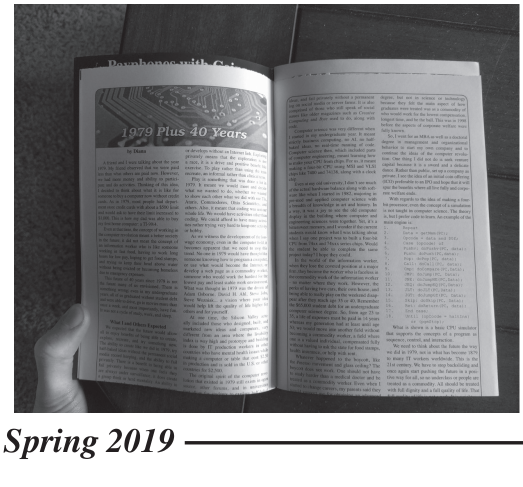
Neil

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

by Diana

A friend and I were talking about the year 1979. My friend observed that we were paid less than what others are paid now. However, we had more money and ability to participate and do activities. Thinking of this idea, I decided to think about what it is like for someone to buy a computer now without credit cards. As in 1979, most people had department store credit cards with about a $500 limit and would ask to have their limit increased to $1,000. This is how my dad was able to buy my first home computer: a TI-99/4.

Even at that time, the concept of working in the computer revolution meant a better society in the future; it did not mean the concept of an information worker who is like someone working in fast food, having to work long hours for low pay, hoping to get food stamps, and trying to keep their head above water without being evicted or becoming homeless due to emergency expenses.

This future of 40 years since 1979 is not the future many of us envisioned. There is something wrong; even in my undergraduate years, all of us graduated without student debt and were able to drive, go to movies more than once a week, and most importantly, have fun. It was not a cycle of study, work, and sleep.

What I and Others Expected

We expected that the future would allow all to enjoy benefits of being able to create, explore, recreate, and try something new. The ability to create like we did in 1979, try half-baked ideas without the permanent social media record keeping, and the ability to fail privately. There is a benefit to being able to fail privately because when one feels they are always under surveillance, it does create a group think or herd mentality. An ability to

or develops without an Internet link. Exploring privately means that the exploration is not a race, it is a drive and positive benefit like exercise. To play rather than using the term recreate, an informal rather than clinical term.

Play is something that was done a lot in 1979. It meant we would meet and decide what we wanted to do, whether we wanted to show each other what we did with our TIs, Ataris, Commodores, Ohio Scientifics, and others. Also, it meant that coding was not our whole life. We would have activities other than coding. We could afford to have many activities rather trying very hard to keep one activity or hobby.

As we witness the development of the low-wage economy, even in the computer field, it becomes apparent that we need to stop this trend. No one in 1979 would have thought like someone knowing how to program a computer, or use what would become the Internet, or develop a web page as a commodity worker, someone who would work the hardest for the lowest pay and least stable work environment. What was thought in 1979 was the dream of Adam Osborne, David H. Ahl, Steve Jobs, Steve Wozniak... a vision where your idea would help lift the quality of life higher for others and for yourself.

At one time, the Silicon Valley actually included those who designed, built, and marketed new ideas and computers, very different from an area where the livability index is way high and prototype and building is done by IT production workers in other countries who have mental health issues while making a computer or table that cost $2.50 in production and is sold in the U.S. or other countries for $2,500.

The original spirit of the computer revolution that existed in 1979 still exists in open source, other forums, and in universities

ideas, and fail privately without a permanent log on social media or server farms. It is also comprised of those who still speak of social issues like older magazines such as *Creative Computing* and *Byte* used to do, along with code.

Computer science was very different when I started in my undergraduate year. It meant strictly business computing, no AI, no half-baked ideas, no real-time running of code. Computer science then, which included parts of computer engineering, meant learning how to make your CPU from chips. For us, it meant making a four-bit CPU using MSI and VLSI chips like 7400 and 74138, along with a clock chip.

Even at my old university, I don't see much of the actual hardware balance along with software like when I started in 1982, majoring in pre-med and applied computer science with a breadth of knowledge in art and history. In a way, it was a joy to see the old computer display in the building where computer and engineering sciences were together. Yet, it's a bittersweet memory, and I wonder if the current students would know what I was talking about when I say one project was to built a four-bit CPU from 74xx and 74xxx series chips. Would the student be able to complete the same project today? I hope they could.

In the world of the information worker, when they lose the coveted position at a major firm, they become the worker who is faceless in the commodity work of the information worker - no matter where they work. However, the perks of having two cars, their own house, and being able to really play on the weekend disappear after they reach age 35 or 40. Remember the $65,000 student debt for an undergraduate computer science degree. So, from age 23 to 35, a life of expenses must be paid in 14 years whereas my generation had at least until age 50; we would move into another field without becoming a commodity worker, a field where one is a valued individual, compensated fully without having to ask the state for food stamps, health insurance, or help with rent.

Whatever happened to the boycott, like the #metoo movement and glass ceiling? The boycott does not work. One should not have to study harder than a medical doctor and be treated as a commodity worker. Even when I started to change careers, my parents said they

degree, but not in science or technology because they felt the main aspect of how graduates were treated was as a commodity of who would work for the lowest compensation, longest time, and be the ball. This was in 1998 before the aspects of corporate welfare were fully known.

So, I went for an MBA as well as a doctoral degree in management and organizational behavior to start my own company and to continue the ideas of the computer revolution. One thing I did not do is seek venture capital because it is a sword and a delicate dance. Rather than public, set up a company as private. I see the idea of an initial coin offering (ICO) preferable to an IPO and hope that it will spur the benefits where all live fully and corporate welfare ends.

With regards to the idea of making a four-bit processor, even the concept of a simulation is not taught in computer science. The theory is, but I prefer code to learn. An example of the main engine is:

```
    Repeat
    Data = getMem(PC);
    Opcode = data and $0f;
    Case (opcode) of
    Pushv: doPushv(PC, data);
    Push: doPush(PC,data);
    Pop: doPop(PC, data);
    Call: doCall(PC, data);
    Cmp: doCompare(PC,Data);
    JMP: doJump(PC, Data);
    JNE: doJumpNE(PC,Data);
    JEQ: doJumpEQ(PC,Data);
    JLT: doJLT(PC,Data);
    JGT: doJumpGT(PC, Data);
    Skip: doSkip(PC, Data);
    Ret: doReturn(PC, data);
    End case;
    Until (opCocde = haltIns)
       or (pgmStop);
```

What is shown is a basic CPU simulator that supports the concepts of a program as sequence, control, and interaction.

We need to think about the future the way we did in 1979, not in what has become 1879 to many IT workers worldwide. This is the 21st century. We have to stop backsliding and once again start pushing the future in a positive way for all, so no underclass or people are treated as a commodity. All should be treated with full dignity and a full quality of life. That

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 destructive 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.

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.

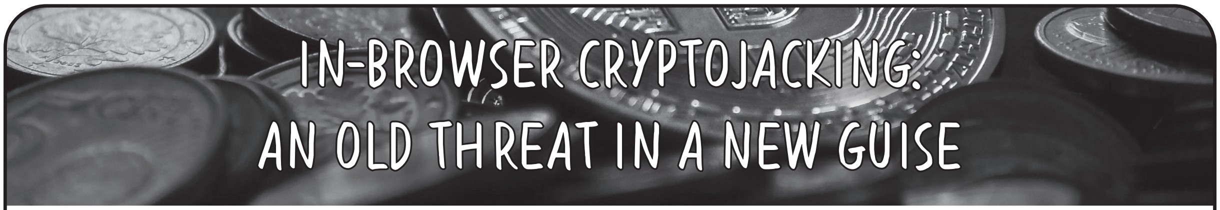
It looks like optical non-linear thin films are the most promising technology for the second generation of quantum computers. These will have to be combined with integrated optical waveguides, photonic crystals, and plasmonic devices to achieve scalable, mass producible quantum computers. These technologies 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

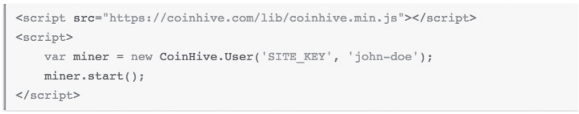
concept of cryptocurrencies to be synonymous and with good reason: it has been one of the most volatile and hence profitable cryptocurrencies in the market, and currently holds the largest well-known market cap for cryptocurrencies, which has brought it immense popularity. But there are a lot more cryptocurrencies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

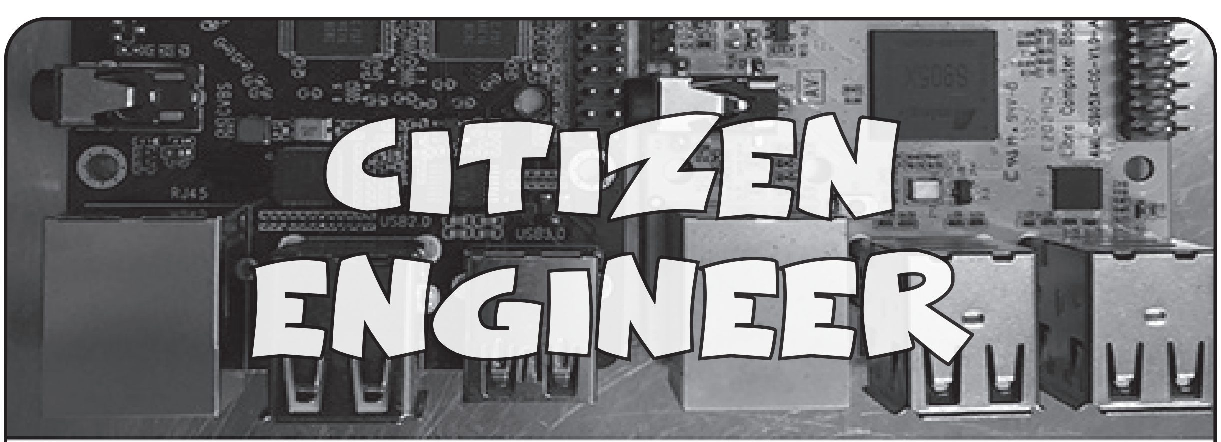
1) Do you get compensated well?

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:

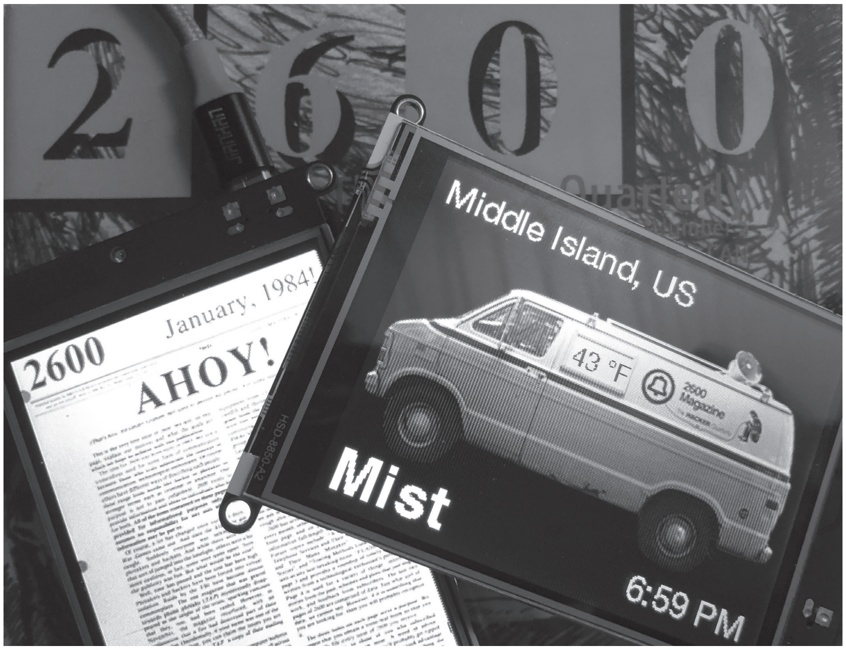
```
>>> j[‘dt’]
1551050040
```

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.

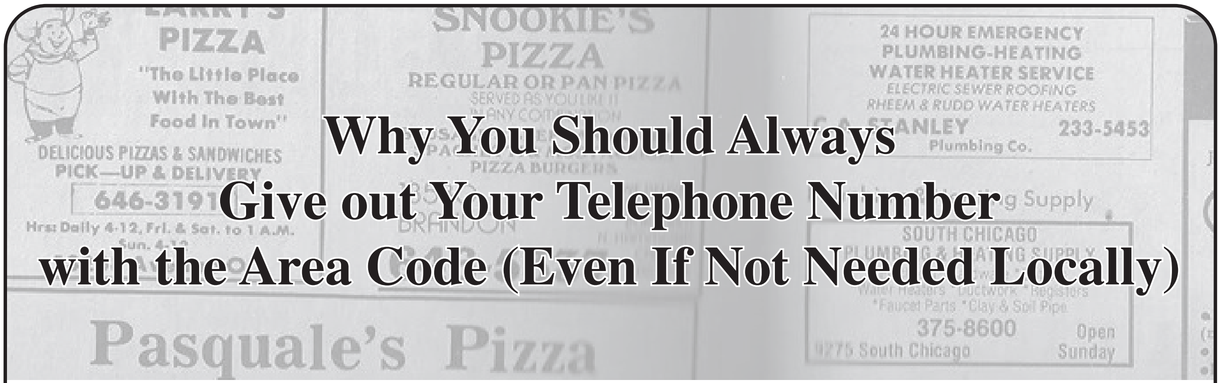
Do you have the local test working locally? Great - though it's usually not smooth to deploy it "in the field." You will want some form of dynamic IP updater in most instances, such as DynDNS or No-IP. You'll need an address to access the Pi from a remote location, one that doesn't change. These will 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.

HACKER HAPPENINGS

Listed here are some upcoming events of interest to hackers. Hacker conferences generally cost under $200 and are open to everyone. Higher prices may apply to the more elaborate events such as outdoor camps. If you know of a conference or event that should be known to the hacker community, ***email us*** at **happenings@2600.com** or by snail mail at **Hacker Happenings, PO Box 99, Middle Island, NY 11953 USA.** We only list events that have a firm date and location, aren't ridiculously expensive, are open to everyone, and welcome the hacker community.

April 19-22
Easterhegg 2019
Technischen Universität
Vienna, Austria
eh19.easterhegg.eu

April 26-28
CarolinaCon 15
Renaissance Charlotte
Charlotte, North Carolina
www.carolinacon.org

May 3-4
THOTCON 0xA
Chicago, Illinois
thotcon.org

May 16-17
Converge
Cobo Hall
Detroit, Michigan
convergeconference.org

May 17-19
Maker Faire Bay Area
San Mateo Event Center
San Mateo, California
www.makerfaire.com

May 17-19
NolaCon
Astor Crowne Plaza
New Orleans, Louisiana
nolacon.com

May 22-23
RVAsec
University Student Commons
Virginia Commonwealth
University
Richmond, Virginia
rvasec.com

May 30 - June 2
GPN19
Karlsruhe University of
Arts and Design
Karlsruhe, Germany
entropia.de/GPN19

May 31 - June 2
CircleCityCon 6.0
The Westin Indianapolis
Indianapolis, Indiana
circlecitycon.com

May 31 - June 2
Hackmeeting 0x16
CSA Next Emerson
Florence, Italy
www.hackmeeting.org/hackit19/

June 21-23
Teardown 2019
Pacific Northwest College of Art
Portland, Oregon
crowdsupply.com/teardown/
➥portland-2019

August 8-11
DEF CON 27
Paris, Bally's, Planet Hollywood
Las Vegas, Nevada
www.defcon.org

August 8-15
BornHack
Hylkedamvej 54
Gelsted, Funen, Denmark
bornhack.dk

August 21-25
Chaos Communication Camp
Ziegeleipark Mildenberg
Zehdenick, Germany
events.ccc.de

September 6-8
DerbyCon 9.0
Marriott Louisville
Louisville, Kentucky
www.derbycon.com

September 13-15
Balkan Computer Congress
Congress Centre
Novi Sad, Serbia
2k19.balccon.org

September 21-22
World Maker Faire New York
New York Hall of Science
Queens, New York
www.makerfaire.com

October 24-25
GrrCON
DeVos Place
Grand Rapids, Michigan
grrcon.org

Please send us your feedback on any events you attend and let us know if they should/should not be listed here.

Marketplace

For Sale

GUIDEBOOK TO COMPUTER AND SMARTPHONE SECURITY by Brandon of Lipani Technologies LLC has been released. This new security book can be purchased at https://leanpub.com/techgeek. Brandon is a certified CompTIA Security+ professional helping users and companies secure their computers, networks, and smartphones across the country. He says, "The purpose of this book is to educate and teach computer and smartphone users about safety and security online."

FOR SALE: ***SAN ANTONIO RADIO MEMORIES*** *- LET 'EM OUT!* Remembering San Antonio Radio in the 40s, 50s, 60s, and 70s. Profits go to ARRL. Visit www.velocepress.com/books/arts/sarm.php to order today!

SECUREMAC.COM is offering popular anti-malware app MacScan 3 to help protect Mac users from malware, spyware, and ransomware. Download a 30-day trial directly from SecureMac.com. Looking for a new podcast? Check out *The Checklist* by SecureMac on iTunes, Pandora, and Spotify.

OPEN SOURCE HARDWARE: crowdfunded and in-stock on Crowd Supply (crowdsupply.com). Includes software-defined radios (SDRs), DIY computers, NASs, FPGA boards, open silicon (RISC-V), hardware encryption/security devices, kite-balloons, workbench tools, optical decoders, and opportunities to help fight the DMCA (see bunnie huang's NeTV2 project).

HACKER WAREHOUSE is your one stop shop for hacking equipment. We understand the importance of tools and gear which is why we carry only the highest quality gear from the best brands in the industry. From WiFi Hacking to Hardware Hacking to Lock Picks, we carry equipment that all hackers need. Check us out at HackerWarehouse.com.

PORTABLE PENETRATOR. Find WPA WPA2 WPS WiFi Keys Pen Testing Software. Vulnerability Scanning & Assessment Customize reports use for consulting. Coupon code 20% off: 2600. https://shop.secpoint.com

HACKERSTICKERS.COM now carries cDc merchandise, sells lock pick sets, Bawls energy mints, and an awesome lineup of hacker clothing including the new Johnny Cupcakes x HackerStickers collaboration Hacker Big Kid Shirt. Get all the goods at HackerStickers.com.

HEATHKIT BOOK: Interested in vintage electronics? *Classic Heathkit Electronic Test Equipment* by Jeff Tranter covers Heathkit's test equipment line, with in depth coverage of different models including oscilloscopes, meters, tube testers, etc., as well as a history of Heathkit and resources for collecting and restoration. 140 pages in 11 chapters plus appendices. Retails for $19.95 from lulu.com and amazon.com.

DEFEND YOUR WI-FI. Coaxifi delivers Wi-Fi over your home's coaxial cabling to eliminate dead zones. Reuse your existing router to send Wi-Fi farther. Check out our new spiral coiled Ethernet cables! 10% off with promo code "SUP2600". coaxifi.com

CLUB-MATE is now easy to get in the United States! The caffeinated German beverage is a huge hit at any hacker gathering. Available in two quantities: $36.99 per 12 pack or $53.99 per 18 pack of half liter bottles plus shipping. Write to contact@club-mate.us or order directly from store.2600.com.

Help Wanted

JOIN THE HTTPS://CODEFOR.CASH community and earn money with freelance programming jobs. All hats welcome!

HOW CAN WE ENJOY OUR PRIVACY when everything has a GPS tracking device attached to it? We want the Big Brothers to stop tracking us everywhere we go. We shall disarm all GPS systems from all of our toys. We must learn how to disconnect the GPS devices through our brothers and sisters in the hacker world, whether we are amateur or professional hackers. We must regain our privacy. Is there a way that we can disarm the GPS system without destroying or harming our merchandise (toys)? Seeking assistant on the GPS network. All are welcome to directly write to me: Mu'mit Muhammad, PO Box 945, Marienville, PA 16239.

Announcements

OFF THE HOOK is the weekly one hour hacker radio show presented Wednesday nights at 7:00 pm ET on WBAI 99.5 FM in New York City. You can also tune in over the net at www.2600.com/offthehook. Archives of all shows dating back to 1988 can be found at the *2600* site in mp3 format! Your feedback on the program is always welcome at oth@2600.com.

COVERTACTIONS.COM is the most comprehensive directory of encryption products anywhere. Search by type, hardware/software, country, open source, platform, and more. Now over 1036 products listed which include 221 VPN's, 192 messaging and 117 file encryption apps. These are just a few of the 28 categories available. There is no faster and easier way to find the encryption product that meets your requirements. Suggestions and feedback welcome. Now featuring news on important encryption issues.

Services

GET YOUR HAM RADIO LICENSE! KB6NU's "No Nonsense" study guides make it easy to get your Technician Class amateur radio license or upgrade to General Class or Extra Class. They clearly and succinctly explain the concepts, while at the same time give you the answers to all of the questions on the test. The PDF version of the Technician Class study guide is free, but there is a small charge for the other versions. All of the e-book versions are available from www.kb6nu.com/study-guides/. Paperback versions are available from Amazon. E-mail cwgeek@kb6nu.com for more information.

UNIX SHELL ACCOUNTS & HOSTING SINCE 1999. JEAH.NET is one of the oldest and most trusted for fast, stable shell accounts. We include hundreds of funny, relevant vhosts for IRC, and access to new and classic *nix programs and compilers. JEAH.NET proudly hosts eggdrop bots, bouncers, IRCD, and web sites w/SQL. *2600* readers' setup fees are waived. BTW: FYNE.COM offers free DNS hosting and WHOIS privacy for $5 with all domains registered or transferred in!

DIGITAL FORENSICS FOR THE DEFENSE! Sensei Enterprises believes in the Constitutional right to a zealous defense, and backs up that belief by providing the highest quality digital forensics and electronic evidence support for criminal defense attorneys. Sensei's digital forensic examiners hold the prestigious CISSP, CCE, CEH, and EnCE certifications. Our veteran experts are cool under fire in a courtroom - and their forensic skills are impeccable. We recover data nationwide from many sources, including computers, external media, tablets, and smartphones. We handle a wide range of cases, including hacking, child pornography possession/distribution, solicitation of minors, theft of proprietary data, data breaches, interception of electronic communications, identity theft, rape, murder, embezzlement, wire fraud, racketeering, espionage, cyber harassment, cyber abuse, terrorism, and more. Our principals are co-authors of *Locked Down: Practical Information Security for Lawyers*, *2nd edition* (American Bar Association 2016), *Encryption Made Simple for Lawyers* (American Bar Association 2015), and hundreds of articles on digital forensics and an award-winning blog on electronic evidence. They lecture throughout North America and have been interviewed by ABC, NBC, CBS, CNN, Reuters, many newspapers, and even Oprah Winfrey's *O* magazine. For more information, call us at 703.359.0700 or email us at sensei@senseient.com.

SQUIDIX provides serious discounts for fantastic web hosting for *2600* readers. We love our clients and they love us. Our *2600* promotion will give you a Super Squid hosting platform for only $26.00 for the first year, then only $9.95 per month when paid annually. Sign up today and get free domain or domain renewal. This offer valid for any new accounts in 2018 and includes a free CPanel transfer of one existing site. Sign up at www.squidix.com

INTELLIGENT HACKERS UNIX SHELL: Reverse.Net is owned and operated by intelligent hackers. We believe every user has the right to online security and privacy. In today's hostile anti-hacker atmosphere, intelligent hackers require the need for a secure place to work, compile, and explore without big-brother looking over their shoulder. Hosted in Chicago with Filtered DoS Protection. Multiple Dual Core FreeBSD servers. Affordable pricing from $5/month, with a money back guarantee. Lifetime 26% discount for *2600* readers. Coupon Code: Save2600. http://www.reverse.net/

ASPIRING TO BE THE MOST ETHICAL TECH SHOP IN THE WORLD, Technoethical.com offers the largest catalog of hardware products certified by the Free Software Foundation (FSF) to Respect Your Freedom (RYF) [fsf.org/resources/hw/endorsement/technoethical]. As a user of Technoethical devices, you have the maximum control over your computing, being able to use, copy, modify, and distribute all the bits in the operating system and, when possible, even at lower levels, such as the boot firmware. The shop sells laptops and servers pre-installed with a fully free (as in freedom) BIOS replacement and GNU/Linux-libre distributions verified and endorsed by the FSF. All x86_64 devices serviced and sold have Intel's intentional backdoor, the Management Engine [u.fsf.org/2g0], completely removed. As the only shop that sells phones with Replicant [replicant.us] pre-installed, you can be the first hacker on your block to own an Android-based device with an operating system that can be compiled completely from source with no proprietary blobs. You can also buy from Technoethical a diverse array of WiFi adapters that work with drivers and firmware that are fully hackable and operate also in the Access Point mode. Moreover, Technoethical provides installation/liberation services for all computers that are also sold as products. You can ship your compatible computer to Technoethical, or ask the team to organize a workshop in your local hackerspace or free software event. With 4 years of experience on the market, Technoethical is operated by a geographically distributed team of hackers from North America, the European Union, Russia, and Australia that closely follow the software freedom principles of the GNU project. Use the coupon code 2600MAG to receive a 5% discount on all Technoethical products. Order today and join Richard Stallman among the many happy customers of Technoethical!

HAVE YOU SEEN THE *2600* STORE? Plenty of features, hacker stuff, and all sorts of possibilities. We accept Bitcoin and Google Wallet, along with the usual credit cards and PayPal. We have an increasing amount of digital download capability for the magazine and for HOPE videos. Best of all, we've lowered prices on much of our stock. Won't you pay us a visit? store.2600.com

LOCKPICKING101.COM - a locksport community driven by lock picking hobbyists and locksmiths alike. New to lock picking or want to advance your skills or help others learn? Just head over to LockPicking101.com and say Mr. Picks sent you!

DOUBLEHOP.ME is an edgy VPN startup aiming to rock the boat with double VPN hops and encrypted multi-datacenter interconnects. We enable clients to VPN to country A, and transparently exit country B. Increase your privacy with multiple legal jurisdictions and leave your traditional VPN behind! We don't keep logs, so there's no way for us to cooperate with LEOs, even if we felt compelled to. We accept Bitcoin and offer automated order processing! Use promo code COSBYSWEATER2600 for 50% off (https://www.doublehop.me).

ANTIQUE COMPUTERS. From Altos to Zorba and everything in between - Apple, Commodore, DEC, IBM, MITS, Xerox... vintagecomputer.net is full of classic computer hardware restoration information, links, tons of photos, video, document scans, and how-to articles. A place for preserving historical computers, maintaining working machines, running a library of hard-to-find documentation, magazines, SIG materials, BBS disks, manuals, and brochures from the 1950s through the early WWW era. http://www.vintagecomputer.net

SKEPTICAL OF GITHUB? sr.ht is an in-progress software suite for hosting open source projects that's more in tune with the hacker way. sr.ht is more modular and more flexible, with features like mailing list driven development and full virt build automation with KVM. Interested in helping test the beta? Reach out to SirCmpwn: sir@cmpwn.com

Personals

PENPALS: Seeking tech or hacker friends to write. I'm 29 years old and from Cleveland, Ohio, but I'm stuck in prison for now. Before the Feds kidnapped me, I worked network operations for an ISP. Being locked up for over 3 years now, I'm out of touch with current technology. There's no Internet here and hardly any resources to keep up. I'm only interested in friends to correspond with and help pass the time - not interested in any criminal/shady activity. Ask me about all the crazy stuff that happens in prison. My other interests include: electronics, renewable energy, open-source projects, antique electronics, homesteading, general aviation, health/fitness, snowboarding, travel & foreign cultures/languages, etc. Do NOT use address labels or stickers; it will be rejected/returned. Looking forward to your letters: Dan Nieberding 61030-060, Federal Correctional Institution, PO Box 1000, Cresson, PA 16630, United States of America.

ONLY SUBSCRIBERS CAN ADVERTISE IN *2600!* Don't even think about trying to take out an ad unless you subscribe! All ads are free and there is no amount of money we will accept for a non-subscriber ad. We hope that's clear. Of course, we reserve the right to pass judgment on your ad and not print it if it's amazingly stupid or has nothing at all to do with the hacker world. We make no guarantee as to the honesty, righteousness, sanity, etc. of the people advertising here. Contact them at your peril. All submissions are for ONE ISSUE ONLY! If you want to run your ad more than once you must resubmit it each time. Don't expect us to run more than one ad for you in a single issue either. Include your address label/envelope or a photocopy so we know you're a subscriber. If you're an electronic subscriber, please send us a copy of your subscription receipt. Send your ad to *2600* Marketplace, PO Box 99, Middle Island, NY 11953. You can also email your ads to marketplace@2600.com.

Deadline for Summer issue: 5/21/19.

THE CIRCLE OF HOPE **VIDEOS**

There's no way you could have seen it all,
whether you were there or not. We can help.

As is our tradition, we have an archive of all of the talks that were given at this year's HOPE conference. There are far too many to list here, but suffice to say, we have more content than ever before - all in high quality HD recordings. The efforts of the folks over at the Internet Society and our amazing A/V team make all of this archiving possible.

We're making these available in three ways:

- Full sets of all talks in MP4 format, no DRM, easy to copy, for $89 on more than one thumb drive (we're at that awkward stage somewhere between 128GB and 256GB).
- On DVD, where a full set of over 100 DVDs now costs only $99 (previously $249) or 99 cents per DVD (previously $2.99). Find a full listing at **xii.hope.net** or on our store.
- For download directly from **store.2600.com** at 59 cents a talk - you get the same MP4s that would come on the thumb drives, but you can choose the ones you want and not have to deal with any hardware.

Looking for HOPE shirts? Sorry, we sold out at the conference for the first time ever!
But we have plenty of other cool things like our 2019 Hacker Calendar,
2600 baseball caps, hoodies, etc., etc.

Want to Become a Digital Subscriber to *2600*?

In addition to the good old-fashioned paper version, you can now subscribe in more parts of the world than ever via Google Play, the Nook, and the Kindle. We're also constantly increasing our digital library of back issues and Hacker Digests.

Head to digital.2600.com for the latest

The Lifetime *Hacker Digest* PDF Subscription

We now have more than 30 years of *2600* digitized with more being added every three months! By becoming a lifetime subscriber, you'll get all of our existing *Hacker Digests*, plus a newly archived one every quarter, along with a brand new digest once a year for as long as you or we are around. $260 gets it all. (Existing lifetime subscribers to the analog edition can get all of this for only $100.)

Visit store.2600.com and click on Downloads/PDF

2600, PO Box 752, Middle Island, NY 11953 USA

"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.

STAFF

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Layout and Design
Skram

Cover
Dabu Ch'wald

Office Manager
Tampruf

Infrastructure
flyko

Network Operations
phiber

Broadcast Coordinator
Juintz

IRC Admins
beave, koz, r0d3nt

Inspirational Music: John Giorno, Cisco Houston, The Only Ones, Stevie Wonder, Chemical Brothers, Coil, The Cranberries, Noah Reid

Shout Outs: Rose Apothecary, Sally & Odd, Ilhan Omar

RIP: Opportunity

***2600* is written by members of the global hacker community. You can be a part of this by sending your submissions to articles@2600.com or the postal address below.**

***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**

MEETINGS

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 Crafty Squire, 127 Russell 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 Centre, food court by HMV.

New Brunswick

Moncton: Champlain Mall food court, near KFC. 7 pm

Newfoundland

St. John's: Memorial University Center food court (in front of the Dairy Queen).

Ontario

Ottawa: World Exchange Plaza, 111 Albert St, 2nd floor. 6:30 pm

Toronto: Free Times Cafe, College and Spadina.

Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

CHINA

Hong Kong: Pacific Coffee in Festival Walk, Kowloon Tong. 7 pm

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

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, Podkopayevskiy Pereulok, 7. 7 pm

Murmansk: Teplo, Teatralny Bulvar, 6**. 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: Beehive Inn on Grassmarket. 6 pm

Glasgow: Starbucks, 9 Exchange Pl. 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: Lux Central, 4400 N Central Ave. 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

Petaluma: Starbucks, 125 Petaluma Blvd N. 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

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: Crescent Coffee Company**,** 311 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

Grand Rapids: Schmohz Brewing, 2600 Patterson Ave SE. 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 Jersey

Somerville: Dragonfly Cafe, 14 E Main St**.**

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. 7 pm

North Dakota

Fargo: West Acres Mall food court.

Ohio

Cincinnati: Hive13, 2929 Spring Grove Ave. 7 pm

Cleveland (Warrensville Heights): Panera Bread, 4103 Richmond Rd.

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.

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, 500 Interstate Blvd S #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

Lexington: Collaboratory**,** 18 East Nelson St, #103. 6 pm

Reston: Refraction, 11911 Freedom Dr. 8th Fl. 7 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.

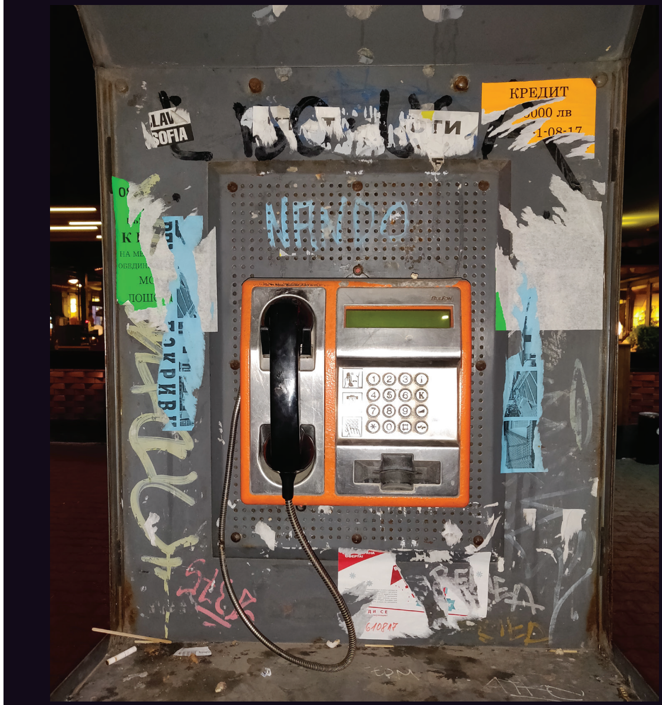
URUGUAY

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.

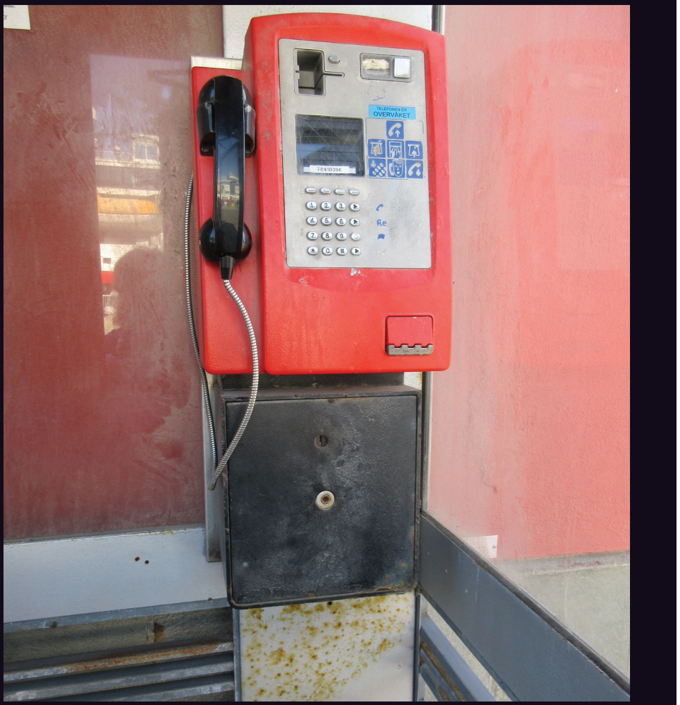
Follow @2600Meetings on Twitter and let us know your meeting's Twitter handle!

Interesting International Payphones



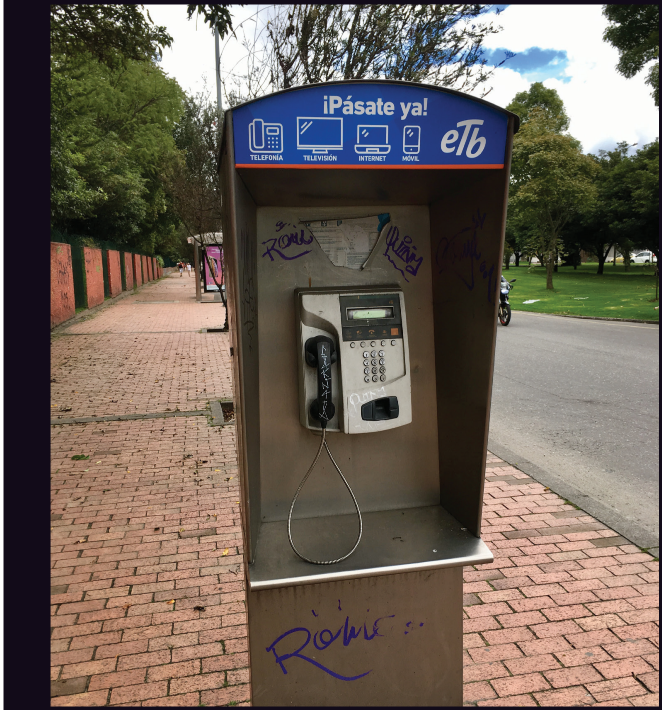
Bulgaria. A common sight in Sofia, and a card-operated phone that looks like it's seen a lot over the years.

Photo by ryoki007



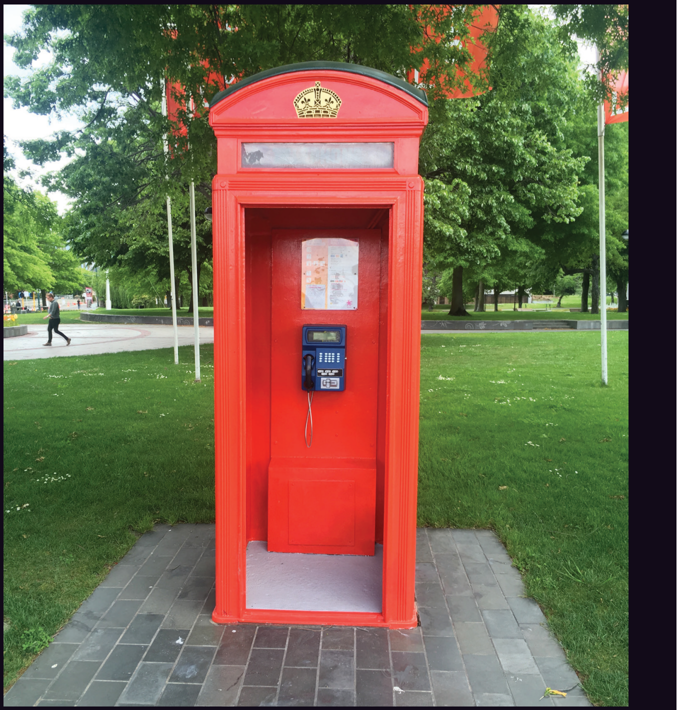
Norway. Possibly the northernmost phone booth in the world, seen in Hammerfest. It's also the only phone booth around.

Photo by Bridget Weller



Colombia. A typical street phone in Bogota, operated by ETB, one of the main telecommunication companies in the country.

Photo by briatych

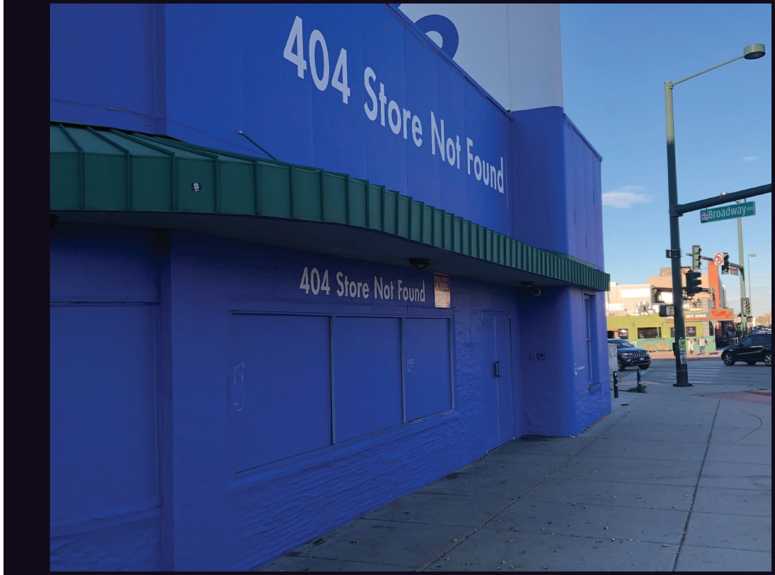


New Zealand. This one wins the prize for the biggest presentation: a pathway, a brilliant shining royal booth, and even some flags in the background. Seen at Victoria Square in Christchurch.

Photo by Declan Maitland

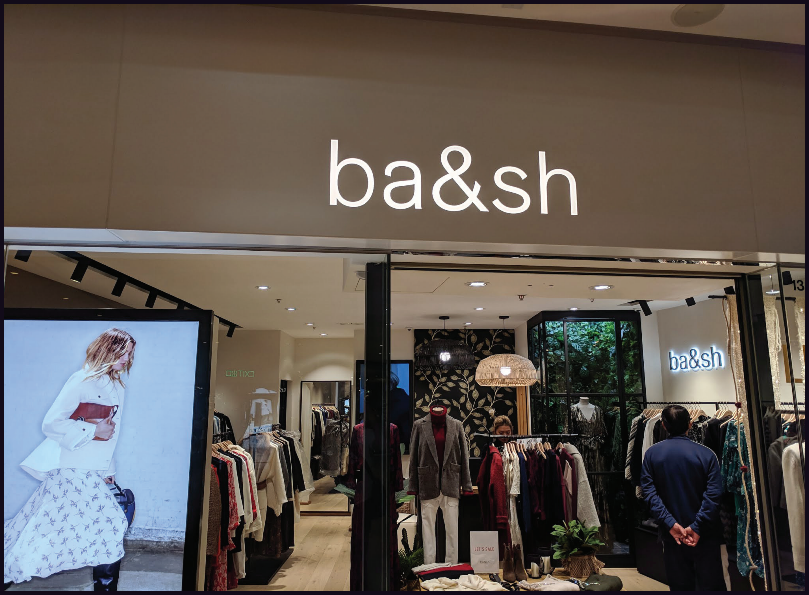
Visit **www.2600.com/payphones** to see our foreign payphone photos!
(Or turn to the inside front cover to see more right now.)

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.

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.



If you've spotted something that has "2600" in it or anything else of interest to the hacker world (such as funny uses of "hacker," "unix," "404," you get the idea...), take a picture and send it on in! Be sure to use the highest quality settings on your camera to increase the odds of it getting printed. Make sure and tell us where you spotted your subject along with any other info that makes it interesting - many photos are eliminated due to lack of detail.

Email your submissions to articles@2600.com or use snail mail to
2600 Editorial Dept., PO Box 99, Middle Island, NY 11953 USA.

If we use your picture, you'll get a free one-year subscription (or back issues) and a *2600* t-shirt of your choice.