

UNREDACTED

MONITORING NEIGHBORHOOD ACTIVITY

Scanner frequencies, trunked systems, and other monitoring hurdles

ADDING KILL SWITCHES TO YOUR PHONE

Securing your phone and its data against theft and attacks

TO PMB OR NOT TO PMB

Answering your questions about private mailboxes



UNREDACTED ISSUE 003

Image: Florian Schmid

IN THIS ISSUE

- 5 From the Editor**
- 6 To PMB or Not to PMB: Your Questions About Private Mailboxes**
- 8 Adding Kill Switches to Your Android Phone**
- 12 The Linux Lifestyle: More Customization and Installation**
- 20 Retrogaming and Privacy**
- 22 Maintaining Multiple Signal Accounts**
- 24 Purging Gmail Data Automatically**
- 26 Exporting Authy 2FA Seed Codes**
- 28 Using a De-Google Pixel**
- 32 The Radio Receiver: Monitoring Neighborhood Activity**
- 35 Mitigating Risk of Online Service Failure**
- 38 A Fresh Look at Standard Notes**
- 40 The OSINT Corner Learning the Linux Command Line**
- 43 HP Dev One with Pop!_OS**
- 46 A FOSS Solution for Receiving Short Code SMS**
- 48 Options for Your Domain**
- 51 Apartment Living**
- 52 Maintaining Privacy in the United Kingdom**
- 54 The Case for Switching to Qubes OS**
- 57 Reader Q&A**
- 59 Book Updates**
- 60 Letters**
- 63 An Anonymous Cross-Country Road Trip**
- 65 Privacy-themed Puzzles**
- 66 Chuckles**
- 67 Final Thoughts**
- 67 Affiliate links**

UNREDACTED is published free of any charge to the reader, and this file may be publicly shared in its entirety. All issues are available for free download at UNREDACTEDmagazine.com. Contact details are also available at this site.

The contents of this publication are copyright © 2022 by UNREDACTEDmagazine.com, and are published via a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International license. You may share it for free as long as you keep the entire file intact. Commercial use is prohibited.

Cover Design by Anonymous Reader. Layout by [Astropost](#). Special thanks to everyone who helped make this happen. You know who you are.



FROM THE EDITOR

By Michael Bazzell

Well, we made it to the third issue. This is by far the largest issue with the most content. I am relieved that the privacy, security, and OSINT communities have joined in this endeavor. Based on temporary cPanel logs, we now estimate the following download stats for the previous issues.

Issue 001 May 1, 2022 through May 31, 2022: **57,317** downloads

Issue 001 June 1, 2022 through June 30, 2022: **9,851** downloads

Issue 002 June 1, 2022 through June 30, 2022): **61,938** downloads

This tells me two things. First, people are still learning about the magazine and then downloading previous issues. Second, the monthly audience of new readers is growing. These are both good things. I plan to watch these stats and provide updates as we go. My goal is to reach the same amount of readers which listen to the podcast.

Whenever I execute any new project, partnership, or business, I always establish a specific plan and stick to it. With this magazine, the plan was "three issues, in three months, at any cost". My staff was given a general direction, and

was allowed to do whatever it takes to get three quality issues made. I believe we accomplished that with this issue. The plan called for a re-evaluation after the third issue was complete, so here we are.

The issues have definitely improved each time, both in terms of content and overall appearance. We welcomed new sponsors which helped justify the work. However, I severely underestimated the effort, time, and resources required to create and publish a monthly magazine. Because of this, we will be moving to a quarterly publication beginning with issue 004 on October 1, 2022. The release schedule will be as follows.

Q1: January

Q2: April

Q3: July

Q4: October

This will allow my staff to work on things at a slower pace so they can keep up with their priorities on the business side of the house. A quarterly publication allows us to take more time with each issue, offer a larger publication, and eliminate the constant deadline burden which we felt with the monthly option. I think this will be better for us and the magazine.

In this issue, I am the author of eight articles, which are spread throughout the magazine. This is not to hoard pages, but simply to provide updates. I have always envisioned this magazine to also serve as a free public platform for necessary updates and revisions to my books. The articles presented this month would have otherwise been reserved for future editions of Extreme Privacy or Open Source Intelligence Techniques. Instead, I offer them here now. Future issues will not likely possess as many articles and updates by me, but the avenue will exist when needed.

This means that future issues will rely more heavily on the community. Please send your best submissions for potential publication. We will always give priority to articles which are tutorial-driven and teach something new. More details can be found at UNREDACTEDmagazine.com.

I sincerely thank my magazine staff (Tyler and Laura), cover designer (anonymous), layout artist (Nick), proofreaders (anonymous), and all contributors. This would be a short text file without you.

MB



Image: Bernard Hermant

TO PMB OR NOT TO PMB: YOUR QUESTIONS ABOUT PRIVATE MAILBOXES

By Michael Bazzell

When I wrote the first edition of Extreme Privacy, I began the conversation about Private Mailboxes (PMBs, which are also commonly called Personal Mailboxes) and their benefits for privacy seekers. I did not receive much feedback about the tactics throughout the first two editions. The third and fourth editions have increased in popularity, and there are now many people embracing many of my ideas. This presents more questions from readers. Lately, over 20% of the questions we receive are centered around PMBs and their proper usage. I present this article as a summary of the concerns we are hearing. All of the questions here were recreated from

the numerous PMB-related emails we received throughout June. First, I should explain the PMB.

A PMB is much more than a simple PO Box address. It provides you a mailing address which is often accepted by institutions that otherwise block CMRA and PO Box addresses. It also allows the collection of mail and distribution to a second address of your choosing. It is basically your new permanent personal address for any mail delivered in your real name. A PMB is a staple for every client. It is also a vital step toward advanced privacy techniques such as obtaining proper vehicle registrations, driver's licenses, passports, and other identification documents each displaying this private address. My

book Extreme Privacy presents many pages of ways to encompass a PMB into your life. Let's now focus on the questions and feedback from this portion of the book.

Q: What differentiates a PMB from a UPS Store or similar option?

A: UPS Stores have a benefit to privacy seekers, but it is not a true PMB. PMB providers exist to collect your mail, scan the outer envelope to notify you of received mail, and offer to ship all of your mail to another location. You might possess a PMB address which you will never physically visit. A PMB address is intended to be your publicly-available address which people and businesses will share with others. It may exist in a different state.

Q: Is a PMB a CMRA?

A: Yes, absolutely. Any type of third-party mail collection service is likely identified as a Commercial Mail Receiving Agency (CMRA).

Q: If a PMB is a CMRA, why is it more special than any other PO Box?

A: A PMB possesses a unique street address and allows you to use "PMB", "Box", "suite", "number", or practically any other identifier to receive your mail. A PO Box requires "PO Box" to be listed on any incoming mail. PMBs are much more forgiving.

Q: Will a bank or utility be fooled if I put "Apt. 143" on my PMB address?

A: No, never. Any business or government entity will absolutely know that your PMB address is a CMRA, regardless of how it is labeled.

Q: So what is the main benefit of the PMB in regard to utilities or a driver's license?

A: If you live in a state which offers "nomad" options, such as Texas, you can easily apply your PMB address on your driver's license. When you need to show any type of identification, it will not contain your true home address. Many PMB providers cater to nomadic people who live in RVs full time. Utilities, insurance providers, and other invasive companies often allow the use of a PMB address, but not a PO Box or UPS store.

Q: Can I purchase a PMB in any state?

A: That is up to you. I prefer PMBs within nomad-friendly states such as Florida, South Dakota, and Texas. Many providers who identify as a PMB will not allow you to use that address on a license and will not re-ship your mail to another location. I also insist on a scanning option so I know what mail is waiting for me.

Q: What is the difference between a digital mailbox (ipostal1.com) vs. a mail forwarding service like Americas Mailbox (americasmmailbox.com) and**how do you tell if a service offers a PMB that can be utilized on a DL, bank accounts, credit cards...etc.?**

A: iPostal is a mail scanning service. It is the technology which services such as Americas Mailbox use to let you know about received mail. Any respectable PMB provider will have documents on their site if their address is allowed to be used on a driver's license.

Q. When shopping for a PMB, what characteristics unique to PMBs do I ask the company about to be able to determine whether the company is a PMB or a regular CMRA (Commercial Mail Receiving Agency)?

A: There are no official characteristics of any PMB (which are all CMRAs) which will force a company to accept it as your home address. Instead, history of use will be a greater benefit. The more items you place in the address of your PMB, the faster you will create publicly-available history of the association. If your bank purchases a consumer report associated with you, and there is no reference to your PMB address, you might be declined. If that report displays your license, vehicle, and insurance under the PMB address, you will have a better chance. Some banks will refuse a NEW account addressed to a PMB, but they will allow you to CHANGE your address associated with an established account. If you are planning a move, generate any financial accounts first, then change your address after 30 days.

Q: I tried to create a business checking account with a PMB address but was declined. How do I convince the bank to allow the address?

A: You may not be able to. I would start with a local bank where you can apply in person. Offer your identification, business papers, EIN documentation, and anything else which displays your PMB address. The more, the better. If desperate, provide a physical address of a hotel where you are staying that evening. Surprisingly, banks have little issue with that, but a proven PMB is suspicious.

Q: When establishing residency in a new state on a long-term basis (with little to no travel, and NOT as a nomad), many states will demand two proof of residency documents with a current local address to get a new license. Given that a local CMRA or out of state PMB would likely be rejected, what would be the best address to give and acquire proof for without revealing a true long-term home address?

A: This is why I prefer to establish residency in nomad-friendly states, but I respect that most readers will not have that luxury. Every state will be unique and there is no clear answer here which will help everyone. However, I can share successes I have had with clients. One client moved to a new state. Instead of waiting until she purchased or rented a home to inform the state of her arrival, we immediately executed a new license for her. At the time, she was staying at a long-term hotel, and we had plenty of documentation to prove that. The DMV accepted a hotel address, and we broke no laws. In order to remain somewhat legal, she stays there on occasion and still possesses that address on the DL. Another client rented a "tiny home" Sprinter van and purchased a 30-day campsite pass. This satisfied the DMV as she was technically living at the campsite with no other home associated with her name. In both scenarios, driver's licenses were obtained within states which would eventually be the true domicile of the client. This makes things more "clean" from a legal, vehicle, and taxation view.

A PMB address alone is not a perfect privacy solution. It is simply one layer toward a full privacy lifestyle. It takes time to establish a history with that address, and you should never expect companies, such as financial institutions, to immediately accept it as a primary address. You will never convince companies that your PMB is your official home residence. That was never the purpose. It is instead a tool in your toolbox. It took me over a year to become established under my PMB. Today, I use no other address for anything associated with my name. ■



Image: Sten Ritterfeld

ADDING KILL SWITCHES TO YOUR ANDROID PHONE

By stateworld

There are many scenarios where having a kill switch or panic button on your electronic devices can come in handy. In this article, I will guide you through the process of setting up kill switches with custom triggers and actions on your Android phone, so you can keep your data safe in the event it gets into the hands of the wrong person. By the end of this article, you will have learned how to set up the following kill switches:

- Lock the device when it is shaken (useful if you drop your phone or if it gets snatched from you while you're using it).
- Wipe the device after too many failed unlock attempts.
- Wipe the device when it has not been unlocked for N days.
- Lock the device when airplane mode gets enabled (useful if someone has access to your unlocked phone, and they enable airplane mode to avoid being tracked or to prevent a remote kill switch from deleting data or locking them out).

- Lock the device when a USB cable is connected.
- Display a notification alert when a wrong PIN has been entered, but only after the device has been unlocked.

The first three of these will be very easy to set up as we will simply be using open-source apps from F-Droid that are specifically made for their respective purposes. The latter three will require a bit more work as we will be creating them from scratch, which also means that you will be able to tweak them however you like. For example, instead of simply locking the device when triggered you can force a shutdown, execute custom code, send an SMS or email alert, or play "Rick Astley – Never Gonna Give You Up" on max volume. Or all of them at once. The possibilities are endless.

Lock the device when it is shaken

Let's start simple. There is a very useful open-source application on F-Droid whose sole purpose is just this. It's called "Private Lock" and can be found at: <https://f-droid.org/packages/com.wesaphzt.privatelock/>. It has a very simple interface and allows you to configure the sensitivity of the

lock trigger to your liking. I highly recommend this app for anyone who sees the benefit in this feature.

Wipe the device after too many failed unlock attempts

Back in April, GrapheneOS [tweeted](#) the following:

We've made significant progress on implementing initial duress features including a wipe PIN/passphrase which could eventually be expanded with a well-designed / written approach to wiping a smaller subset of data in a rigorous way. We're taking great care to do it properly.

This is great news for the GrapheneOS users, but these features have yet to be implemented, and we don't know when they will be. Besides, not everyone is running GrapheneOS. In any case, we can use the open-source application "Wasted" to achieve somewhat similar functionality. This app won't let us configure a wipe PIN, but it will allow us to erase all data on the phone after a certain amount of failed unlock attempts. It's pretty straightforward to use, so I won't bore you with instructions. Download Wasted here: <https://f-droid.org/packages/me.lucky.wasted/>.

Full disclosure: I have not personally tested if the wipe feature actually works. Do not rely on this app without first testing it on your device.

Wipe the device when it has not been unlocked for N days

The Wasted app discussed above also provides another very useful feature, namely an “inactivity wipe” feature, which can be configured to automatically wipe your device if it is not unlocked within a certain amount of time. By default it is set to 7 days, but if you press and hold the “Wipe on inactivity” text you will be presented with the following options:

- 1 day
- 2 days
- 3 days
- 5 days
- 7 days
- 10 days
- 15 days
- 30 days

Wasted also lets you connect it to a remote trigger/panic button such as [Ripple](#), which allows you to trigger a wipe remotely. Again, do not rely on this app without first testing it on your own device.

Creating your own kill switches with custom triggers and actions

The kill switches presented above are very easy to set up and install, and while they are definitely useful, you may desire some additional privacy and security measures. I will now show you how to create the following kill switches completely from scratch:

- Lock the device when airplane mode gets enabled.
- Lock the device when a USB cable is connected.
- Display a notification when a wrong PIN has been entered, but only after the device has been unlocked.

The application we will be using to

create these kill switches is not open source, though this guide can likely be adapted to work with open source software with relative ease. Alternative FOSS applications you can experiment with will be provided toward the end of the article.

First, download the “Automate” app from Aurora Store here: <https://play.google.com/store/apps/details?id=com.llamalab.automate>. This application allows you to automate virtually any task on your Android device. We will be using it to “automate” a device lock when certain conditions are met. Before you install it on your phone, I encourage you to read the privacy policy at: <https://llamalab.com/automate/doc/privacy.html>. To summarize, it essentially states that they do not collect or share any personal or sensitive data, nor do they collect any usage statistics (analytics). I do not have any issues with their privacy policy, though, just to be safe, I have disabled its network permission (along with any other not strictly necessary permissions) on my GrapheneOS device.

Lock the device when airplane mode gets enabled

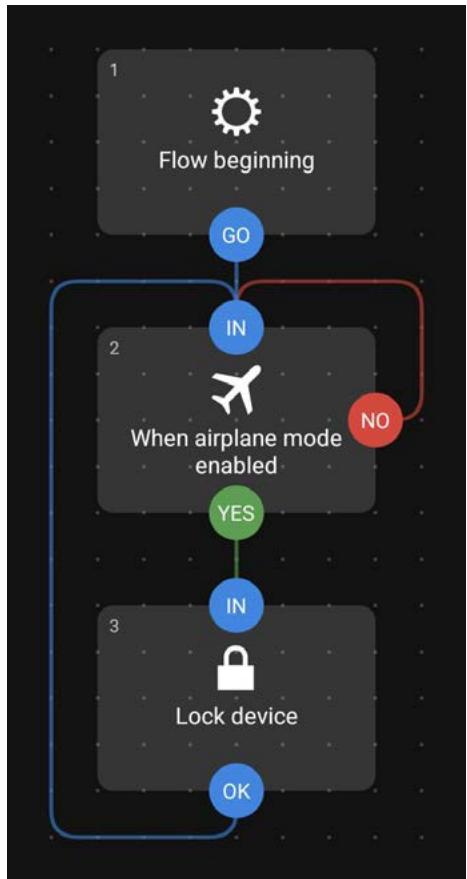
The first custom kill switch we will be creating is one that detects when airplane mode gets enabled, and then immediately locks the device. After the device has been locked it will wait for airplane mode to be disabled, and then start over from the beginning.

- Inside Automate, create a new automation “flow” by pressing the blue “+” button in the lower right-hand corner.
- Press the “+” button again, then navigate to “Connectivity” and select “Airplane mode enabled?”. This will create a block which detects, like the name suggests, when airplane mode gets enabled.
- Drag the “GO” output from the “Flow beginning” block to the “IN” of the airplane mode block to connect them.
- Add a “Device lock” block which can be found under “Interface”.

- Drag the “YES” output from the airplane mode block to the “IN” field of the device lock block. This will lock the device when airplane mode gets enabled.
- Now we need to loop the flow since it stops right after locking the device. To do this, first connect the “OK” output from the device lock block to the “IN” of the airplane mode block, and then drag the “NO” output from the airplane mode block to its own “IN” field. This will start the flow over from the beginning when airplane mode gets disabled.

Note that the airplane mode blocks don’t constantly check to see whether airplane mode is on or off, they only detect the action of enabling and disabling airplane mode. This means that the phone is still able to be unlocked while airplane mode is enabled.

If you followed all the steps correctly, your flow should look like this:



Lock the device when a USB cable is connected

For our second custom kill switch, we want to trigger a device lock when a power source is detected (in other words, when a USB cable is connected). Like the previous flow, this will also not prevent your phone from being unlocked while it is connected to a USB cable, only the initial cable connection will trigger a device lock.

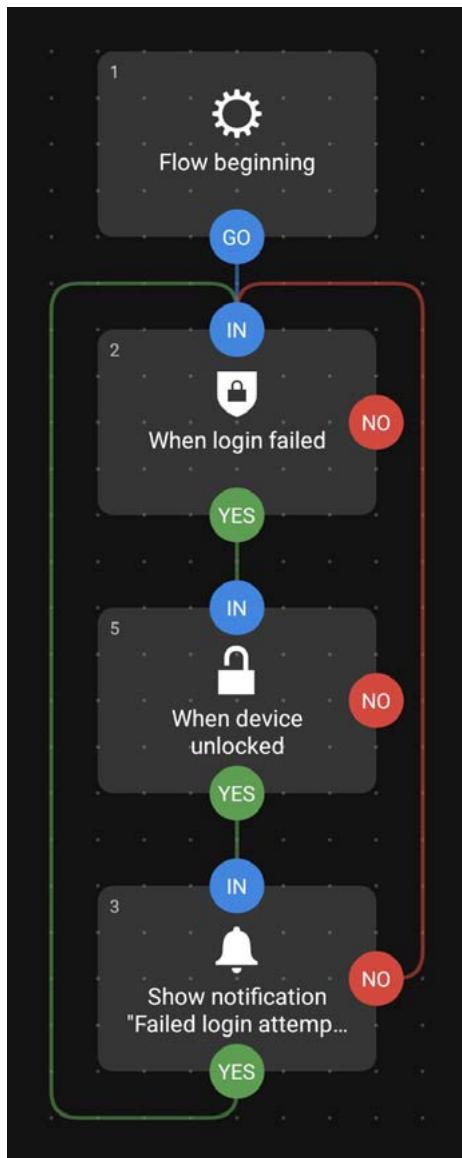
Since this flow actually works the exact same as the previous one, we can simply duplicate the flow we just made. The only thing we need to do now is replace the airplane mode block with a "Power source plugged?" block which can be found under "Battery & power".

Display a notification alert when a wrong PIN has been entered, but only after the device has been unlocked

In this flow, we first want it to wait for a failed unlock attempt. Once that has been detected, it should wait for the device to be successfully unlocked, and then display a notification.

- Add a "Login failed?" block and connect it to the flow beginning.
- Connect the "YES" output to a "Device unlocked?" block.
- Connect the "YES" output from that block to a "Notification show" block.
- Click on the notification block to customize the icon, text and behavior of the notification to your liking. I like enabling the "Cancellable" option which makes the notification dismissible. You can play around with the settings to see what works best for you.
- Connect both the "YES" and "NO" outputs from the notification block to the "IN" of the login failed block. This will loop the flow regardless of whether you click on the notification or dismiss it.

The finished flow should look as such:



Final notes

In Automate, when you have confirmed that your flows work as intended, make sure to enable the "Run on system startup" option in the settings to make your flows automatically start after booting your phone. Always disable this option while working on your flows, as having it enabled during testing can get you stuck in an infinite loop. Not fun, trust me...

Speaking of loops, a loophole to get around the "30 running blocks" limitation on the free edition of Automate is to clone the app to your work profile and run two instances of the application simultaneously. This will effectively give you 60 running blocks which should be more than enough. It can also be a good idea to turn off logging on your flows to save space on your phone.

I hope you have found some of the things presented in this article interesting and/or useful, and are able to apply some of these techniques to your own situation. ■

Alternative open source apps

I tried playing around with some of these applications but I didn't particularly enjoy the experience, so I ultimately went with Automate. Not only does Automate have a LOT more features than these FOSS alternatives, it's also much more polished and easier to use. However, you may prefer one of these options for your own setup:

Easer: <https://f-droid.org/packages/ryey.easer/>

Easer (beta): <https://f-droid.org/packages/ryey.easer.beta/>

Automation: <https://f-droid.org/packages/com.jens.automation2/>



The World's Only All-in-One Privacy App



Sudos act as digital firewalls to eliminate data trails.

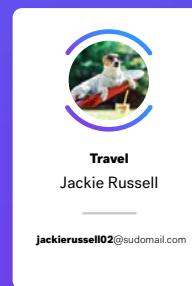
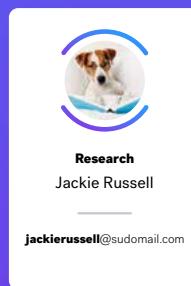


Call, Text, Email, Browse, Shop and Pay

Privately and Securely

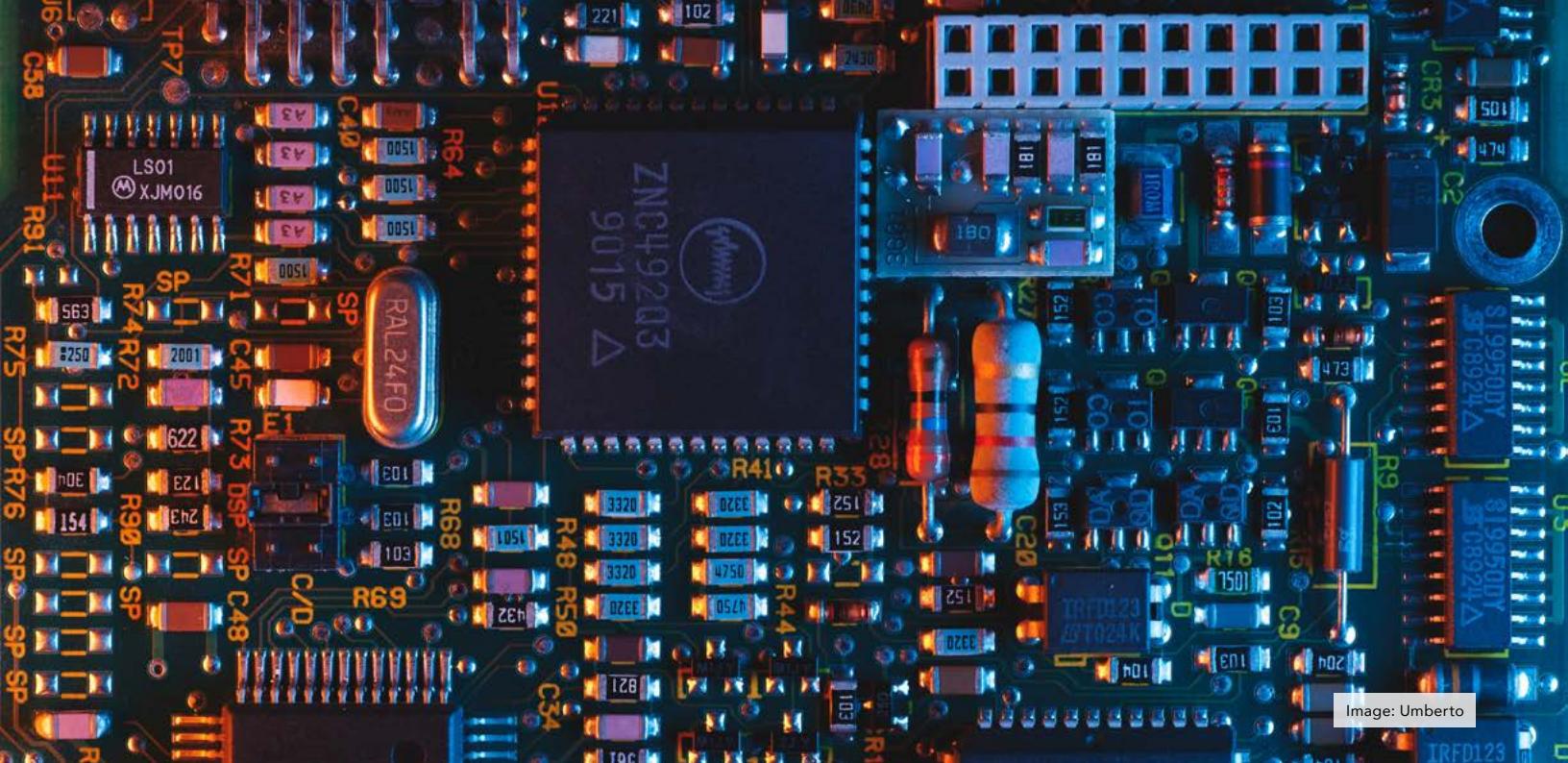


Create digital identities, or **Sudos**, for different situations.



Sign up without an email, phone number or password | MySudo.com/bazzell





THE LINUX LIFESTYLE: MORE CUSTOMIZATION AND INSTALLATION

By Michael Bazzell

The Linux Lifestyle is a monthly column all about Linux. From new useful apps to working through Linux frustrations, this section aims to introduce others to a more secure operating system.

Last month, I began this series with an introduction to the Linux operating system Pop!_OS which I have installed on my primary computer, a System76 Thelio desktop. I discussed my reasons for using Linux, choosing System76, transitioning to Pop!_OS, configuring NextDNS, and hardening Firefox. It was the primer for this series and gets us to this second article. If you have not read issue 002, please start there. This month, I want to tackle the remaining steps which I take as part of every new Linux installation.

The first item I want to tackle is Firefox. Last month, I explained how Flatpak offered multiple versions of

Firefox which were each isolated from one another. Several readers reported they could not see such options while others had no issue. I was able to reinstall Pop!_OS and confirm I could no longer access these features from a native machine, but could within a VM. I do not know why this is. It might be another custom setting I am carrying over. Fortunately, we have other options. The following steps within Terminal download the beta version of Firefox, extract the files, move them to the appropriate location, and link them for easy access.

```
sudo ln -s /opt/firefox-beta/
firefox /usr/local/bin/
firefox-beta

nano ~/.local/share/
applications/firefox-beta.
desktop

[Desktop Entry]

Name=Firefox Beta
Comment=Web Browser
Exec=/opt/firefox-beta/firefox
%u
Terminal=false
Type=Application
Icon=/opt/firefox-beta/
browser/chrome/icons/default/
```

```
sudo ln -s /opt/firefox-beta/
firefox /usr/local/bin/
firefox-beta
```

Next, we must create a desktop shortcut. Execute the following line into Terminal and then paste the entire text which immediately follows into the black screen.

```
default128.png
```

```
Categories=Network;WebBrowser;
```

Pressing “ctrl-o” saves the data, “enter” confirms the file name, and “ctrl-x” exits the screen. Entering “rm firefox.tar.bz2” into Terminal deletes the downloaded file. You should now have a new Firefox Beta icon in your applications menu which launches independent of your standard Firefox installation. Next, let’s install the Developer version of Firefox in Terminal.

```
wget -O firefox.tar.bz2  
'https://download.mozilla.  
org/?product=firefox-  
devedition-latest-ssl&os=  
linux64&lang=en-US'
```

```
tar xvf firefox.tar.bz2
```

```
mv firefox firefox-dev
```

```
sudo mv firefox-dev /opt
```

```
sudo ln -s /opt/firefox-dev/  
firefox /usr/local/bin/  
firefox-dev
```

```
nano ~/.local/share/  
applications/firefox-dev.  
desktop
```

```
[Desktop Entry]
```

```
Name=Firefox Dev
```

```
Comment=Web Browser
```

```
Exec=/opt/firefox-dev/  
firefox %u
```

```
Terminal=false
```

```
Type=Application
```

```
Icon=/opt/firefox-dev/browser/  
chrome/icons/default/  
default128.png
```

```
Categories=Network;WebBrowser;
```

```
ctrl-o
```

```
enter
```

```
ctrl-x
```

```
rm firefox.tar.bz2
```

The following repeats the process and installs the Nightly edition.

```
wget -O firefox.tar.bz2  
'https://download.mozilla.  
org/?product=firefox-  
nightly-latest-ssl&os=linux-  
64&lang=en-US'
```

```
tar xvf firefox.tar.bz2
```

```
mv firefox firefox-nightly
```

```
sudo mv firefox-nightly /opt
```

```
sudo ln -s /opt/firefox-  
nightly/firefox /usr/local/  
bin/firefox-nightly
```

```
nano ~/.local/share/  
applications/firefox-nightly.  
desktop
```

```
[Desktop Entry]
```

```
Name=Firefox Nightly
```

```
Comment=Web Browser
```

```
Exec=/opt/firefox-nightly/  
firefox %u
```

```
Terminal=false
```

```
Type=Application
```

```
Icon=/opt/firefox-nightly/  
browser/chrome/icons/default/  
default128.png
```

```
Categories=Network;WebBrowser;
```

```
ctrl-o
```

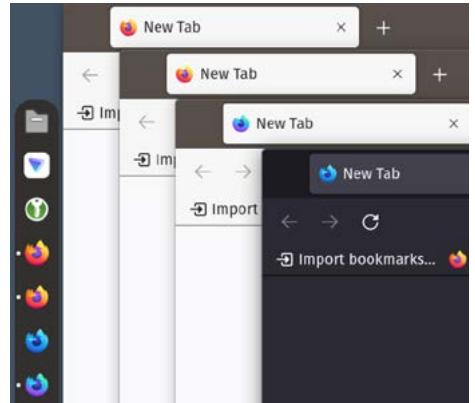
```
enter
```

```
ctrl-x
```

```
rm firefox.tar.bz2
```

You should now have four independent versions of Firefox for various tasks. I use the stable version as my daily default hardened browser, and it sees most activity. It possesses any custom extensions or add-ons desired. The beta version is used for Google Voice number management as explained in issue 002, the dev version is strictly hardened for sensitive tasks, and the nightly is kept “loose” and default for use with web calls and other annoyances which can be a hurdle

when privacy modifications are applied. There are no extensions installed within the non-stable options. The following image displays my options in the dock, each executed to the right. Note that you will need to manually update these new versions on occasion within Menu > Help > About Firefox.



When I convert a client from Windows or macOS to Linux, I must provide alternative applications and services which replace their standard invasive counterparts. While I mentioned a few staple applications which I install to every Linux machine I use, I did not offer any explanation about those selections. Let’s dig into that now. The following are my preferences for open-source Linux applications, presented alphabetically, and their role in my usage.

- **Atom:** Atom is a text editor with many enhancements over alternative options. I use this to create and maintain my websites. I also rely on it for editing text files and scripts.
- **Audacity:** This replaces other audio recording software such as Adobe Audition or Garage Band. It is not as robust as the closed-source proprietary options, but it gets the job done. I prefer the simple interface, and have used it for every podcast I have ever recorded.
- **BleachBit:** This replaces CCleaner as a cleanup utility. It removes cache files and most unnecessary clutter left behind within the operating system. I launch it weekly and select every option except the full disk wiping feature.

- **Calibre:** This software organizes all of my e-books, PDFs, magazines, and other text files. It connects to practically any e-reader device to upload and manage all content. It replaces any Kindle software or service.
- **Deja Dup Backups:** The minimal and efficient Pop!_OS installation does not include any default backup solution. I prefer to install Deja Dup Backups and manually execute it weekly. I accept the default Home directory backup, choose “Local Folder”, select a location on an external drive, password protect the backup, and allow it to proceed. I then update the backup weekly.
- **EasyTag:** If you manage your own music collection locally, you will need a good MP3 tag editor. None of them on Linux will ever truly replicate the options on macOS or Windows, but EasyTag comes the closest.
- **Electrum:** This is a Bitcoin wallet which does not require any registration to third-party exchanges.
- **Element:** This is a client for Matrix which I prefer to the browser-based version. It stores my credentials and allows an easy connection. I use this for secure communication to members of the online video training.
- **Etcher:** This application allows me to easily create bootable USB drives of various operating systems.
- **Firefox:** Pop!_OS includes an APT version of Firefox. I install the additional versions previously explained in order to possess multiple isolated Firefox browsers.
- **FreeFileSync:** I use this program for synchronizing my documents to external VeraCrypt containers. Since I store much of my personal data within containers outside of my home directory on my Linux machine, that data is not backed up with Deja Dup Backups. FreeFileSync can also synchronize data to another machine in my home, such as my media server. I use this for all local off-site backups of all data. Deja Dup Backups is to archive system settings and configurations. FreeFileSync is to copy my personal data.
- **GIMP:** This is the best Photoshop replacement for Linux.
- **GnuCash:** This software organizes my personal and business finances. It allows import of common financial statement files without invasive connections to third-party data services.
- **Handbrake:** This is the only software you will ever need to rip DVDs, render movie files, or compress videos.
- **KeePassXC:** This is the only password manager I use. It is completely offline.
- **Kodi:** This is my media center. It hosts and organizes all of my audio and video media stored within my system. I also possess an independent media center with the same setup.
- **Linphone:** I own many VoIP telephone numbers and Linphone allows me to place or receive calls from all of them.
- **MailSpring:** MailSpring is an email client which I use after switching from Thunderbird. It allows me to backup all of my email via IMAP in order to possess my own offline copy. I execute this weekly to download all communications from the week. If you use Proton Mail, you will also need to install the Proton Mail Bridge application. Be sure to choose the offline “Desktop” version of MailSpring to avoid using their paid email hosting service.
- **NewsFlash:** This is my preferred RSS reader. I configure RSS feeds of all desired news sources, Twitter feeds, etc., and allow it to retrieve only the text content.
- **OnlyOffice:** Pop!_OS includes LibreOffice as a Microsoft Office replacement, but I prefer OnlyOffice. It is open-source and can open Microsoft Word files better than LibreOffice Writer. It also has an appearance closer to Microsoft Word and includes an Excel and PowerPoint replacement.
- **Proton Mail Bridge:** This application allows me to synchronize my Proton Mail account(s) to the MailSpring email client.
- **Proton VPN:** This application configures a system-wide VPN. Note that I only install this on my laptop, since my desktop is always behind my home firewall VPN. However, having it available on all machines allows easy transition to different servers.
- **Signal:** This is a secure E2EE messenger.
- **Standard Notes:** This is my desired note-taking application. See my review of their paid options later in this issue.
- **Tor Browser:** If you ever need to access Tor websites, you will need Tor Browser.
- **Transmission:** If I ever need to download a torrent, this is my preferred option.
- **VeraCrypt:** This is a secure file container creation application which encrypts your data.
- **VirtualBox:** This is open-source virtual machine software which allows us to launch contained versions of other Linux or Windows operating systems.
- **VLC:** This is a media player which supports a large number of audio and video codecs.
- **Wire:** This is a secure E2EE messenger.

You could install all of these options within the Pop!_OS Shop, but I never do that. Instead, I create a single command which installs all of these

applications within Terminal. Before I present the commands, let's understand our options. Within Terminal, you can type the following to search for a

specific application by any term. In this example, I searched for any apps called "Signal".

```
flatpak search signal
```

The first two results are below:

Name	Description	Application ID	Version	Branch	Remotes
Signa...	Private messenger	org.signal.Signal	5.44.1	stable	flathub
Signal	Private messenger fo...	org.signal.Signal	1.28.0	beta	flathub-beta

The Application ID and Remotes data can be used to install the application. The following command would install the stable version of Signal.

```
flatpak install flathub org.signal.Signal
```

If I wanted to see a list of all installed Flatpak applications, I would execute the following.

```
flatpak list
```

Finally, if I wanted to update all Flatpak programs, I would use the following command.

```
flatpak update
```

Why would I do all of this when I could just use the Pop!_OS Shop application? Terminal commands allow me to create a single command which installs all of my desired Flatpak programs at once. This allows me to easily replicate my installation on another computer or when I want to reformat my device. The following single command installs Atom, Audacity, Calibre, Deja Dup Backups, EasyTag, Electrum, Element, FreeFileSync, GIMP, GnuCash, Handbrake, KeePassXC, Kodi, MailSpring, NewsFlash, OnlyOffice, Proton Mail Bridge, Signal, Standard Notes, Tor Browser, Transmission, VLC, and Wire. I have included all of these commands within the "Commands" file available on the downloads page at UNREDACTEDmagazine.com.

```
flatpak install flathub io.atom.Atom org.audacityteam.Audacity com.calibre_ebook.calibre org.gnome.DejaDup org.gnome.EasyTAG org.electrum.electrum im.riot.Riot org.freefilesync.FreeFileSync org.gimp.GIMP org.gnucash.GnuCash fr.handbrake.ghb.org.keepassxc.KeePassXC tv.kodi.Kodi com.getmailspring.Mailspring com.gitlab.newsflash.org.onlyoffice.desktopeditors ch.protonmail.protonmail-bridge org.signal.Signal org.standardnotes.standardnotes com.github.micahflee.torbrowser-launcher com.transmissionbt.Transmission org.videolan.VLC com.wire.WireDesktop -y
```

The following command installs Android Tools (ADB for interacting with Android devices), BleachBit, FDUPES (removes duplicate files), ffmpeg (required by many video tools), Linphone, Nautilus Admin (adds right-click option to edit a system file as admin), RipGrep (searches through large data sets), VirtualBox, and yt-dlp (downloads online videos):

```
sudo apt update && sudo apt install android-tools-adb bleachbit fdupes ffmpeg linphone nautilus-admin ripgrep virtualbox yt-dlp -y
```

VeraCrypt presents a unique situation for installation, but we can accomplish

it with the following commands.

```
wget https://launchpad.net/veracrypt/trunk/1.25.9/+download/veracrypt-1.25.9-Ubuntu-22.04-amd64.deb
```

```
sudo apt install ./veracrypt-1.25.9-Ubuntu-22.04-amd64.deb -y
```

```
rm veracrypt-1.25.9-Ubuntu-22.04-amd64.deb
```

If you want to install Etcher, conduct the following.

```
curl -1sLf \'https://dl.cloudsmith.io/public/balena/etcher/setup.deb.sh\' | sudo -E bash
```

```
sudo apt update
```

```
sudo apt install balena-etcher-electron -y
```

If you are a Proton VPN subscriber, conduct the following to install their app for VPN protection.

```
wget https://protonvpn.com/download/protonvpn-stable-release_1.0.1-1_all.deb
```

```
sudo dpkg -i protonvpn-stable-release_1.0.1-1_all.deb
```

```
sudo apt update
```

```
sudo apt install protonvpn -y
```



system7₆

THELIO DESKTOP COMPUTERS

- Crafted with Intention in Denver, Colorado



WELCOME TO THELIO

The Thelio desktop line was born from the belief that humans are capable of anything. We believed that we could make an open hardware desktop that's powerful, compact, quiet, beautiful, upgradeable, backed by lifetime support, and manufactured in the United States—so we did. Powered by either AMD or Intel with a top of the line components, the Thelio is made to help you unleash your potential. What will you make with it?

LEARN MORE: [HTTPS://S76.CO/SYSTEM76THELIO](https://s76.co/system76thelio)

I realize this may seem like a lot of work, but it actually simplifies everything. You could now copy these commands and paste them into Terminal for seamless installation of all desired apps. If you need to repeat this process, you have the same commands ready for you. This is how I keep all of my Linux machines identical. My System76 laptop is an exact replica of my Thelio desktop. This way I am familiar with my system while I am traveling. I can copy all of the commands presented at <https://unredactedmagazine.com/data/003.txt> in one step and walk away. In a few minutes, my desired apps are installed. Next, let's tackle custom settings. The following command displays most available settings within your operating system.

```
gsettings list-recursively
```

The following commands allow us to make system changes directly from Terminal.

Remove wallpaper image:

```
gsettings set org.gnome.desktop.background picture-uri ''
```

Set wallpaper to solid color:

```
gsettings set org.gnome.desktop.background primary-color 'rgb(66, 81, 100)'
```

Clear the Dock:

```
gsettings set org.gnome.shell favorite-apps []
```

Place the Dock on the left:

```
gsettings set org.gnome.shell.extensions.dash-to-dock dock-position LEFT
```

Add the custom apps to the Dock:

```
gsettings set org.gnome.shell favorite-apps "[ 'org.gnome.Nautilus.desktop', 'protonvpn.desktop', 'org.keepassxc.KeePassXC.desktop', 'firefox.desktop', 'firefox-beta.desktop', 'firefox-dev.desktop', 'firefox-nightly.desktop', 'org.signal.Signal.desktop', 'com.wire.WireDesktop.desktop', 'im.
```

```
riot.Riot.desktop', 'org.standardnotes.standardnotes.desktop', 'com.getmailspring.Mailspring.desktop', 'com.gitlab.newsflash.desktop', 'io.atom.Atom.desktop', 'org.onlyoffice.desktopeditors.desktop', 'virtualbox.desktop', 'gnucash.desktop', 'org.electrum.electrum.desktop', 'tv.kodi.Kodi.desktop', 'org.freefilesync.FreeFileSync.desktop', 'org.gnome.gedit.desktop', 'org.gnome.Calculator.desktop', 'bleachbit-root.desktop', 'org.gnome.Terminal.desktop', 'gnome-control-center.desktop', 'io.elementary.appcenter.desktop', 'pop-cosmic-applications.desktop']"
```

Decrease the Dock icon size (change the number higher or lower based on your needs):

```
gsettings set org.gnome.shell.extensions.dash-to-dock dash-max-icon-size 20
```

Disable Bluetooth:

```
sudo service bluetooth stop
```

Disable recent file storage:

```
gsettings set org.gnome.desktop.privacy remember-recent-files false
```

Remove temporary files:

```
gsettings set org.gnome.desktop.privacy remove-old-temp-files true
```

Remove old trash files:

```
gsettings set org.gnome.desktop.privacy remove-old-trash-files true
```

Disable automatic screen lock:

```
gsettings set org.gnome.desktop.session idle-delay 0
```

Allow files to be deleted permanently:

```
gsettings set org.gnome.nautilus.preferences show-delete-permanently true
```

Show hidden files:

```
gsettings set org.gnome.nautilus.preferences show-hidden-files true
```

Show hidden files:

```
gsettings set org.gtk.Settings.FileChooser show-hidden true
```

Set "Old" files to one day:

```
gsettings set org.gnome.desktop.privacy old-files-age 1
```

Enable "Do Not Disturb":

```
gsettings set org.gnome.desktop.notifications show-banners false
```

Display battery percentage:

```
gsettings set org.gnome.desktop.interface show-battery-percentage true
```

You could now replicate all of these settings by copying the commands available on the website and pasting them into Terminal. This allows me to make sure all of my machines have identical privacy settings. As a reminder, you can find all of these at <https://unredactedmagazine.com/data/003.txt>. I encourage you to take these resources and create your own custom commands. Once you have that done, you are a few commands away from rebuilding your system or replicating a secondary machine. I believe it is worth the hassle.

Next, I want to configure mobile Android applications to run natively on my Linux desktop. I rely on Anbox for this. Install it with the following steps. Note that it requires the Snap package manager. I am fine with this only for apps that truly require it.

```
sudo apt install snapd -y  
snap install --devmode --beta anbox -y
```

You now have the Anbox application installed, but it may not load (or be visible). Reboot your computer. If

launching the app still stalls on a loading screen, conduct the following.

```
cd ~/Downloads  
git clone https://github.com/  
choff/anbox-modules.git  
cd anbox-modules  
.INSTALL.sh
```

You now have a minimal Android interface available by launching the Anbox application icon. However, we can do better. My primary purpose for this software is to run MySudo within my desktop. Therefore, I install it by navigating to <https://www.apkmirror.com> and searching for MySudo. This connects me to <https://www.apkmirror.com/apk/anonyme-labs-inc/mysudo-private-secure/mysudo-private-secure-1-8-0-release/mysudo-private-secure-1-8-0-4-android-apk-download/download/> and allows me to download the Android APK file for the program. I then conduct the following based on that specific version.

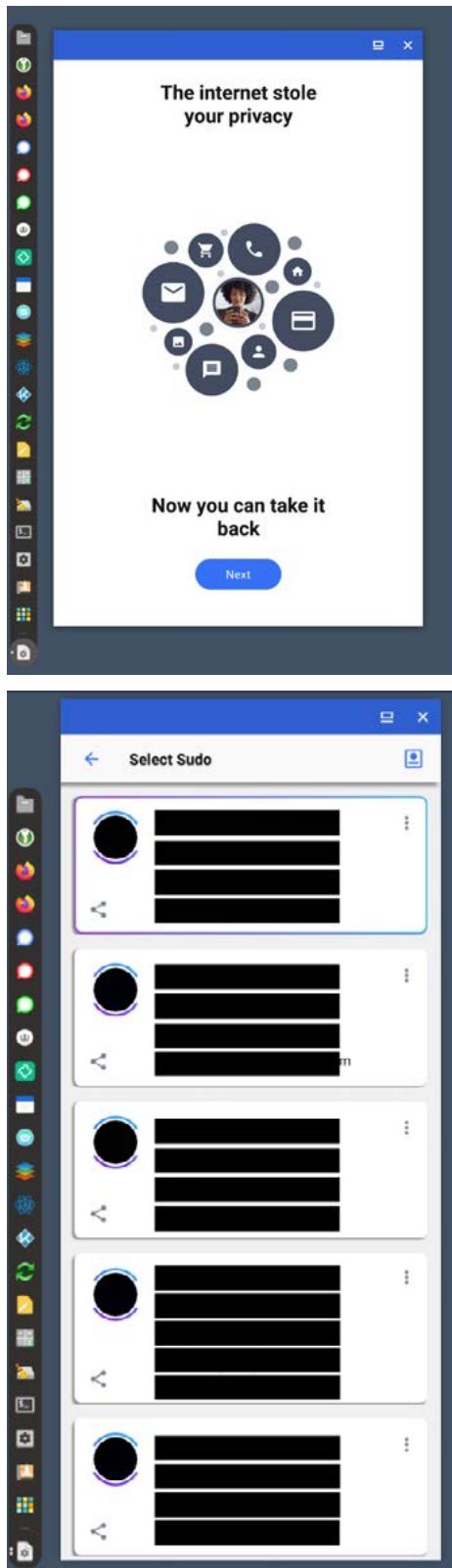
```
cd ~/Downloads  
adb install com.anonyme.  
mysudo_1.8.0-13894_  
minAPI24(x86_64)(nodpi)_  
apkmirror.com
```

You should now see the MySudo application within the Anbox emulator. However, I want a traditional program icon within my applications menu. I executed the following within Terminal.

```
ln -s ~/snap/anbox/common/  
app-data/applications/anbox  
~/.local/share/applications/  
anbox
```

I can now add the MySudo icon to my Dock and execute a traditional mobile application without launching a full emulator such as Genymotion. The following image displays the program along my Dock, ready for me to synchronize my account from my actual mobile device. The second image is a redacted view of my accounts ready for communication. This allows me to

monitor text messages from all nine MySudo numbers.



If your application display is small due to your monitor resolution, the following command will increase the size of any installed Android apps.

```
adb shell wm density 320
```

You could now install mobile versions of Signal, Snapchat, or practically any other "mobile-only" application. Remember this method when I explain multiple Signal numbers later in this issue. Anbox is still in beta. While I had no issues installing this on my Thelio desktop with Pop!_OS, these steps would not work with the same operating system installed within a 2015 MacBook Pro. Hardware can make or break this feature. You may need to do your own troubleshooting online.

Next, we face the issue of encrypted drives. If you followed Pop!_OS's default installation settings, your internal host drive is fully encrypted. If it were stolen or examined, the decryption password would be required to access any data. However, the data on any other drives is easily accessible. In the past, I have encouraged readers to create an encrypted VeraCrypt container on the external drives and store all data securely within it. This is still valid advice, but there are other options. Lately, I have been relying on Linux Unified Key Setup (LUKS) for encryption. From within Pop!_OS or many other Linux builds, open the Disks application, select the target drive, click the settings icon, choose the "Format Partition" option, select the internal Ext4 option, enable the "Password protect volume" feature, and assign a very secure password. The entire external drive is now encrypted. The benefit is that you do not need third-party decryption software to access your data. The risk is that you will need a recent Linux system to unlock your drive. My fellow privacy nerds might use both LUKS and VeraCrypt.

You can also access a disk speed test utility within the Disks application. This is beneficial for those who collect

large data sets, such as breach or leak data, and use Terminal utilities such as RipGrep to search through the data. Knowing the overall speed of your drives may help you decide the best storage location for your data. My 2018 MacBook Pro had a very fast internal drive attached to the motherboard. I could read data from it at 2,500 MBps. This was amazing ... at the time. My new System76 Thelio desktop possesses a NVMe drive which can read data at 7,000 MBps. For comparison, a portable USB spinning-disk drive can probably read at 125 MBps. You can test all of your disks within Linux. Click on them within the Disks application, click the settings icon, choose the "Benchmark Partition" option, select the "Start Benchmark" option and identify the read speeds. This may change your desired data storage setup, especially for breach data. There is much more detail about data breach

queries within my book Open Source Intelligence Techniques.

I think I will wrap this up for the month. Take the time now to document the installations and customizations of your own Linux machine. When you buy a new computer or need to wipe out an existing device, data restoration will be much easier. Configuring my new Thelio desktop took only a few seconds of effort and twenty minutes for all installations to finish. These commands, along with your backups, will make sure you don't spend valuable time configuring your next machine. Oh, and this also applies to all virtual machines. I often spin up a new Pop!_OS VM for testing and quickly possess a fully configured instance. There are a lot of possibilities here. All commands within this article (and issue) can be found at <https://unredactedmagazine.com/data/003.txt>.

Several readers asked for a photo of my daily setup. I assume this is an attempt to obtain evidence which could be scrutinized to identify my location or a point to pivot for a social engineering attack, but I will bite. Below is my new Thelio desktop at my test bench. You may also recognize the Sangean short wave receiver (issue 002) and the Uniden BCD996P2 (later in this issue). Be nice, and ignore my cable management. ■



RETROGAMING AND PRIVACY

By wind

I can remember a sliver of the 80s, but most of my childhood was in the 1990s. Video games were my escape from an often traumatic childhood under the thumb of a religious cult. The internet wasn't accessible to me until the end of the decade in a very constrained form. I understood video games to be a product in the same way as a paper book or CD: one buys it from a store, enjoys unlimited, private use of the product, can lend it to others, and when tired of it can sell it or trade it in.

Sometime around the turn of the millennium, a transition happened to internet-enabled play that much wealthier kids than me with freer access to the internet experienced. Starcraft, Diablo II, Quake, and Warcraft III brought massive popularity to PC gaming and put play against other human beings at the forefront of the experience in a way that was only possible from the same couch on a split screen before. This mostly came from innocent enough intentions, and had straightforward monetization schemes. If one wanted access to the game's servers, some small amount of money would be paid monthly to continue playing, and if unaffordable that month, the single-player version was an option (until the birth of the MMO, that is).

The first moment of self-awareness from the gaming industry about the profitability of exclusive online play and, most importantly, gaming as a service rather than a product came with World of Warcraft, the online game that would reshape gaming culture forever. It wasn't the first MMORPG, but it was the first online game of its kind that achieved the same cultural ubiquity as a Mario or Sonic single-player console game. It was everywhere, and articles

were churned out on mainstream news sources about people giving up their lives to play this game. An unintended side effect of online-only play was the dampening effect it had on the software piracy scene, which had recently sank the Dreamcast console with its freely-copyable game CDs. At the same time, it was beginning to crank out sophisticated emulators like ZSNES which allowed games of relatively recent vintage to be played for free on unspecialized PC hardware.

This bifurcation of online play and emulated play represents the yin and the yang of our focus as the privacy community. Online play represents a host of privacy invasions, especially with the advance of mere "copy protection" (for example, adding an extra security ring to an original PlayStation disc that must be present to read the game) to "Digital Rights Management," or DRM, an online-exclusive software authentication check with the software distributor that allows the game to run, which started to reach prominence in the later part of the first decade of the century during the infancy of platforms like Games for Windows Live and Steam. DRM transformed online functionality, and therefore authentication, from the exclusive concern of online games to the standard practice of games of every kind, including traditional single-player story-driven experiences which required nothing from an internet connection.

Most contemporary online games require a single sign-in to a big tech ecosystem, such as Microsoft, Google, Sony, Valve, Epic, or Facebook. Virtually every click and keystroke is registered and evaluated. The ecosystem's game launcher has low-level access to one's file system and hardware for information scraping. A game which was "bought"

is only owned in the sense that one has a license registered to his account that authorizes the download of the game. It cannot be traded in or lent to a friend (with one artificial scheme in place from Valve as a caveat). Consoles were slower to catch up because game enthusiasts were accustomed to the ability to buy physical games in a store, but as things stand, the vast majority of console games are online-purchased. Those that are bought physically present an intended layer of inconvenience (who wants to get up and grab the disc?) and are often shipped with bugs that require patching the version that came on the disc anyway, mandating sign-in to the walled garden and opening one up to the same privacy violations as online purchases.

By contrast, emulators are overwhelmingly open-source projects, meaning one can view and compile the code personally, and make changes if necessary. The ROMs are typically of older vintage such that they are designed to run on a simple system hooked up via analog video connection to a TV, and so they have no capacity to call home to the parent game company or software ecosystem. The games and save files can be easily backed up to disk, and the emulators provide enhancements the original consoles did not have like savestates and fast-forward. As time has gone on and consoles have taken on more online features, their corresponding emulators have been designed to sandbox those features and can disable them or simulate their connections in a private way. Recent developments have even been made to resurrect long-dead online platforms by the community, such as the original Xbox Live service.

The ethics of software piracy are outside the scope of this discussion,

but if one's conscience is pricked by playing a game he did not buy, there are increasingly accessible and affordable ways to dump the original games onto disk that require very little technical knowledge (see the Retrode device and the Polymega console). These are defined in law as "personal backups" and are explicitly legal.

The future does present problems as even the operating systems that run these emulators become more online-oriented and more designed to value the rights of copyright holders over the privacy of consumers, but if you are reading this publication, you are likely either running a free and open-source operating system like Linux or BSD, or you're well aware of these operating systems and have considered switching. The latest versions of Microsoft Windows and macOS now require an online sign-in to be used – which means that there is a potential permanent association between 'pirated software' (ROMs) and 'piracy contraband' (emulators) with one's real-world identity (Microsoft or Apple account). Care should be taken whenever these associations between activity and identity are present. It only takes one bill to be passed in favor of rights-holders to start crackdowns on these kinds of activities using tools in the operating system already available to tech giants. With the death of Internet Explorer, Microsoft has demonstrated they are ready and willing to remove software from a machine without consent.

I will close out this article with recommendations for those who are new to retrogaming. In my opinion, the greatest device the emulation scene has ever produced is known as the MiSTer FPGA project. This is a project built upon a device called the DE-10 Nano developed by Intel. This device uses an electronic device description language called Verilog that simulates the interaction of multiple hardware components in real time. This allows for transistor-level emulation of consoles, computers, and arcade machines at nearly perfect cycle accuracy, with none of the lag inherent in traditional

PC software emulation. The project has produced dozens and dozens of "cores" as they are called, representing a particular piece of hardware, spanning the time from the dawn of personal computing until the late 1990s. (As of this writing, the 'crown jewel' of the MiSTer platform is the original PlayStation core and its 4,000+ game library of 2D and 3D titles, which could easily occupy one's free time for the remainder of their life.) The device requires modest technical knowledge to set up but can be updated online through a few button clicks and then disconnected. Cores are being added and updated weekly. The cost of entry is fairly expensive, especially during the pandemic's chip shortage, and the DE-10 Nano, USB card, and additional RAM modules can cost \$300-500 depending on configuration.

The RetroArch project is a suite of software emulators that can run on privacy-respecting operating systems such as Linux, BSD, and private flavors of the Android Open-Source Project. What the software gives up in terms of cycle accuracy and input lag it makes up in its \$0 cost of entry on the computer or phone one already owns, as well as its expansive list of emulated platforms, allowing emulation of relatively modern systems (such as the Nintendo 3DS) or older systems that the MiSTer lacks the resources for, namely the Sony PS2 or Nintendo 64/GameCube/Wii. Other standalone emulators of even more recent vintage are being developed and will likely make it to the project in time.

The one online storefront I can hesitantly recommend for game software is GOG. GOG has agreements with many large publishers to sell their older (2010 and earlier) PC games packaged for newer operating systems in a DRM-free form, along with a decent selection of contemporary indie game titles also lacking DRM. Linux support is ample but sporadic, and services like Lutris and Wine can allow games to run that would otherwise be restricted to Windows. Always use discretion when running non-free software without a sandboxed environment, and there

are some publishers, most notoriously Paradox, that publish their games "DRM-free" on GOG only to ask for an online sign-in at startup to access much of the game. Protection tools like AppArmor can be useful to limit network connectivity to games that have no business being online.

Finally, if the reader has a heavy sense of nostalgia and a craving for authenticity, older games and consoles still 'work' in a real sense. An entire online community has been created around the maintenance and enhancement of older consoles and computers. Devices called "flash carts," with the Krikzz Everdrives as the Kleenex of the flash cart market, use FPGA technology to simulate the electronic signal I/O through a traditional game cart system like the Nintendo SNES or Sega Genesis, which allows the loading and playing of ROMs on real hardware. Later CD-based consoles may use similar devices called ODEs (Optical Drive Emulators) that replace the often-failing CD drives in late 1990s and early 2000s consoles. Video modifications for many consoles and analog video scalers allow lag-free HDMI video to connect to modern televisions, replacing increasingly-incompatible and poor quality connections. They also allow online game streaming. The best way to find out information on this hobby is the website and YouTube channel RetroRGB, and the online storefront StoneAgeGamer sells flash carts, modifications, and niche accessories for older consoles. Standard anonymization steps apply when visiting these resources.

If one has adopted a privacy lifestyle and seeks to bring their entertainment habits into line with his philosophy, retrogaming is an inviting hobby. Whatever is lost in terms of graphical fidelity and convenience is easily made up in the freedom gained by the use of open-source software and offline hardware devices.

wind is a software developer, husband, and father of two based in the United States. ■

MAINTAINING MULTIPLE SIGNAL ACCOUNTS

By Michael Bazzell

On my podcast, I have mentioned that I possess several Signal numbers. One I reserved for immediate family and close friends while the others are devoted to clients. This helps me compartmentalize personal life and work. Several listeners have asked how this is possible since Signal only allows one account per device, and their desktop software simply synchronizes to a single authorized account. This article explains my process.

The first step is to create the accounts you need. Each requires a unique telephone number, but any VoIP number should work. I have countless MySudo, Google Voice, and Twilio numbers at my disposal for this purpose. You will need either a mobile device or Android emulator to create the accounts. If you have a GrapheneOS device, you could install Signal, create a new account, synchronize it to your desktop, and then delete the account from the device. This would allow you to start the process over with a new number on mobile while keeping the desktop versions active. However, I do not like this option. It allows Signal to store encrypted data intended for the mobile device which never gets delivered. I prefer a different approach.

In my OSINT book, I explain the methods for installing an Android virtual machine (VM) through VirtualBox or Genymotion. I create a new minimal instance of Android, install Signal via Aurora Store, and create a new account associated with a VoIP number. If I need another Signal account, I replicate the

process with a new VM. I then keep all VMs and occasionally boot them up to download all pending encrypted messages from Signal. This also allows me to re-synchronize the accounts to desktop, if needed.

Once you have multiple Signal accounts created through a mobile emulator, you can associate them to your desktop. If you followed my Linux tutorials with Pop!_OS, you already have your "primary" signal application installed. I use this to connect to my main Signal account used for friends and family. I can now access all messages on either my primary mobile device or the desktop application. Now, I want another version of Signal on my desktop for a client. For this, we will need Snap installed on our desktop OS. Pop!_OS does not include this by default, but it can be easily installed with the following command.

```
sudo apt install snapd
```

You can now install a second copy of the Signal desktop application with the following command.

```
sudo snap install signal-desktop
```

This installs the Snap version, which can exist alongside our primary Flatpak version. Both are isolated from each other and do not share hardware details from our desktop to Signal confirming the association. However, both use the same IP address if you have concerns about that. I do not since I use a VPN being shared with many other customers.

Assume that you want to add a third desktop version of Signal. You cannot install another Flatpak version, and repeating the Snap command above will tell you that Signal already exists. This is where we will use an experimental feature within Snap. The following command enables this option.

```
sudo snap set system experimental.parallel-instances=true
```

Now we can install a second (or third, fourth, etc.) version of Signal with the following commands.

```
sudo snap install signal-desktop signal-desktop_2
```

```
sudo snap install signal-desktop signal-desktop_3
```

```
sudo snap install signal-desktop signal-desktop_4
```

The numbers at the end of the command force Snap to install new isolated versions of the application. After you reboot, you should see evidence of your actions. The following is my Applications menu after installing three copies of Signal to the same machine.



You can now open each of these and pair your accounts from mobile (or emulator) to desktop. The identical names and icon colors confuse me, so I take further action. I keep my primary

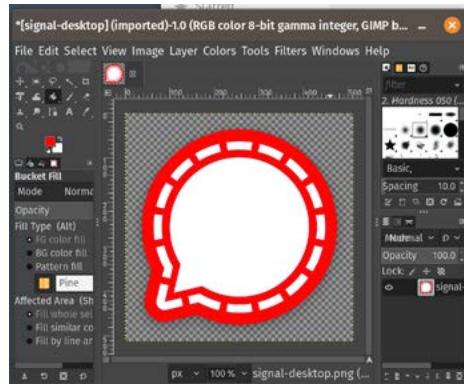
Flatpak installation of Signal as-is, but I want to change the others. If I browse to `/var/lib/snapd/desktop/applications/signal-desktop_desktop.desktop` on my Linux machine, I can open the file as an administrator to make changes (this is why I include that step in the previous Linux article). The icon setting within this file should appear as below.

Icon=`/snap/signal-desktop/386/usr/share/icons/hicolor/512x512/apps/signal-desktop.png`

I navigate to this graphic, right click it, and choose to "Open with other application". I select GIMP so that I can modify this icon. I use the "Bucket Fill Tool", select my desired color, click on blue area of the icon, choose "File" > "Export as", and save my new version to my Documents folder (or any location desired). I then return to my open file located at `/var/lib/snapd/desktop/applications/signal-desktop_desktop.desktop` and change it to my new icon similar to the following.

Icon=`/home/[YourUserName]/Documents/signal-desktop.png`

The following shows my edits within GIMP.



I can now repeat that process to change the other application icons. Below is an example of three installations of Signal with a unique color for each. Blue is friends and family, green is past clients, and red is current high-risk clients. I confess that I actually possess over 40 instances of Signal for various needs.



The image below displays three open instances of Signal ready for pairing to active accounts.



This is likely overkill for most readers, but I encourage you to possess at least two active versions of Signal. It can prevent future abuse of the primary account reserved for a minimal set of people. In my case, it may be the opposite. If my family starts spamming me, I can move over to the client instances for some peace. ■



INDUSTRY LEADING TECHNOLOGY AND A 24X7 SOC WORKING FOR YOU

Cyber threats are evolving rapidly. SMBs and Enterprise businesses are looking to their Managed Service Providers to provide them with cybersecurity solutions. Our managed SOC is highly-skilled in the constantly evolving threat landscape and will provide absolute security for you and your clients.



FORTIFY24X7.COM | (800) 989-2647 | INFO@FORTIFY24X7.COM



PURGING GMAIL DATA AUTOMATICALLY

By [YooperOriginalSince1992](#)

Here is something that I felt would be useful for people transitioning away from Google. I recently went through my old Gmail account to clean it up after setting up the permanent forwarding (and deleting). There are two handy techniques that can be used to make your life easier, one solution makes use of the built-in search tool and the other involves using Google Sheets and Google Apps Script. Overall, Gmail does not make it easy to really do a thorough purge of everything. This is a quick and fairly clean solution. To purge your Gmail account of all mail, conduct the following.

- Search for all 'label:read' and 'label:unread'.
- Select all, which selects all content on the page.
- Click the "Select all XXXX conversations" which will select all search results.
- Click the trash can icon to delete.
- Go to the Trash folder, then select all and again "Select all XXXX conversations".
- Click "Permanently delete" and confirm.

Optionally, go through and delete your custom labels in Settings, although it's purely cosmetic at this point. Delete all the custom filters in Settings since they are not needed anymore. If you've set up Gmail to forward everything to another address and then trash their copy, you will quickly realize that this means that it will still sit in the Trash for 30 days. For some people, this may be acceptable, but for me it seems like a long time for something I don't need anymore, as well as that it shouldn't sit around longer than it's needed. By default, that would mean I would need to keep logging in to clear the Trash bin out. There has to be a better way, and there is.

The solution to this is to use Google Sheets and the built-in Google Apps Script infrastructure. For non-coders, this may seem daunting, but it really isn't. All the code you need can be copy-pasted and is all of a handful of lines. This will take less than 5 minutes to process.

Go to sheets.google.com and generate a new "blank" sheet document. Name it whatever you like (or leave it untitled), but a good name might be, "Gmail Trash Auto-Delete Tool" or the like. Now that you have the document created, we need to attach the code to this "project" that this document will host (for free) for you. At the top of the page, go to the "Extensions" menu and go to "Apps Script". This will generate a new app with an empty function called "myFunction". Select all of the data and delete this function, there should be nothing left. Copy and paste the following code (see image below).

```
function deleteForever() {
  var threads = GmailApp.
    search("in:trash");
  for (var i = 0; i < threads.
    length; i++) {
    Gmail.Users.Messages.re-
    move("me", threads[i].get-
    Id());}}
```

This script searches for all emails that are in the trash, loops through all of the found emails and removes them from the Trash, and is essentially is the same as clicking the "Permanently Delete" button for each message. The "me" specifies that this will only run

on the email account that authorizes this script, so if person123@gmail.com gives permissions to this script, then "me" is the equivalent of person123@gmail.com. The value could specify some other email, but person123@gmail.com then must have permissions to manage it or the script will throw an error. Make sure you save the script.

Now you could run it, but it will fail, because it needs permission to interact with your Gmail account, and for that you have to give it some API and OAuth permissions still. Google Apps Scripts actually makes this really easy. It's all handled in app. Click the plus icon next to "Services" on the left-hand menu. It should bring up a list of Services to add to this app. Scroll down to find "Gmail API", select it and click "Add". Now, you should see "Gmail" listed below "Services" in the left menu.

Finally, you can click the "Run" button. This will say, "This project requires your permission to access your data." Click "Review Permissions". This will ask you choose an already signed-in account or to sign into your account. Follow the sign in instructions (if you aren't already) and click your account you want this script to affect. It will list the permissions required to run the script, which can sound pretty scary, but overall, you "wrote" the code for this app, so you know what this app will be doing; it will be deleting your trash.

As-is, running this script manually is no different (if not more difficult) than going into your email and manually clicking the "Delete Permanently" button, so we need to add some time-based automation to this. On

the far left-side of the screen, there is a "Clock" icon. This is where you can create a "Trigger" to run this script automatically on the hour (or some other interval).

Click the "Go to Dashboard" button (it just closes the dialog).

In the lower right corner, click "Add a Trigger".

Change "Select event source" to "Time-Driven".

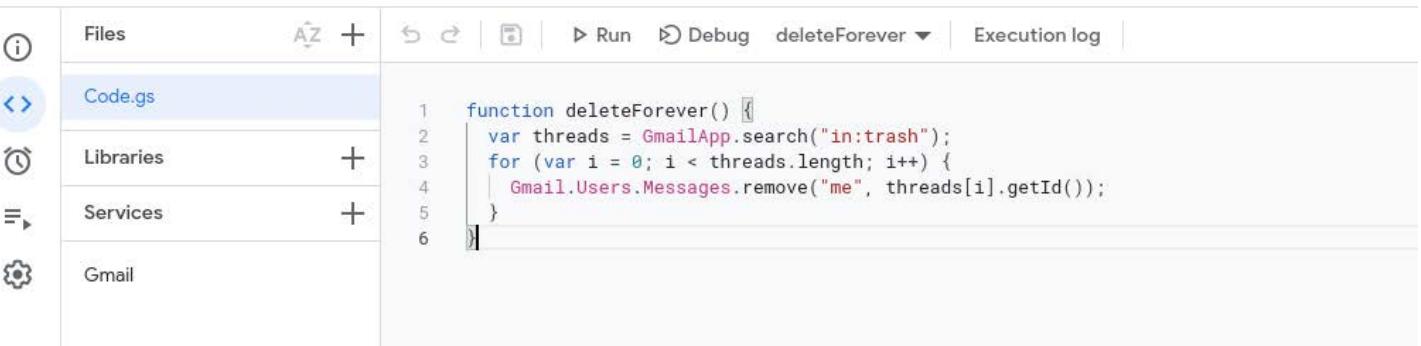
Change "Select type of time based trigger" and the options provided to your desire.

Click "Save".

If you haven't given permissions, it will prompt you at this point. In my case, it actually gave me an error about blocking pop-ups. If you get the same error, just create the same Trigger following this step again and on "save" the second time it seems to create the pop-up to give permissions just fine. If everything saved fine, then you should see a new trigger on the Trigger Dashboard. You can edit this if you want to make changes by going to the far right and clicking the "edit" (pencil) icon. All done!

Now, when your Gmail messages are forwarded, the Trash will be routinely "permanently" deleted much faster than 30 days with no login or extra effort required. When you go to delete/remove all your Google Docs files leave this one so it can keep operating. ■

Apps Script Auto-Delete Gmail



```
function deleteForever() {
  var threads = GmailApp.search("in:trash");
  for (var i = 0; i < threads.length; i++) {
    Gmail.Users.Messages.remove("me", threads[i].getId());
  }
}
```



Image: Brina Blum

EXPORTING AUTHY 2FA SEED CODES

By Michael Bazzell

I have often recommended Authy to my readers and clients. Authy is a 2FA token application which synchronizes your tokens across all platforms. I stand by my recommendation, as Authy has been the easiest option to make sure you always possess your codes on any device. I have noticed that much of the privacy community is now actively irate at Authy. A few YouTube privacy experts have bashed Authy because they do not allow you to download the seed codes which were used to generate your tokens, while other open-source apps allow this. The argument could be made that it is a security benefit that Authy cannot see (therefore share) your seed codes with anyone from

within the app. If I capture your seed codes, then I can recreate every 2FA token for your accounts in real time. That is a huge risk in my eyes. However, that has not stopped the YouTubers from convincing the internet that only open-source solutions which can easily share your seed code should be used. I disagree. Users should instead document their own seed codes during 2FA configuration, and not rely on third-party services to maintain them in a way which is easily accessible.

I never encourage privacy-enthusiasts to rely on Google (YouTube) for all of their privacy advice, but I must weigh in on something specific which keeps popping up. Many of these new experts are falsely claiming that it is impossible

to extract your Authy seed codes, and therefore everyone should generate new codes within a new application and close their Authy accounts. You absolutely can export your Authy codes, and we are going to do it together. It is much simpler than re-doing all of your hard work in the event that you want to switch providers. Many people claim that Authy locks your codes in order to force you to stick with their product. I do not agree with this. I think they do it so that (1) they cannot see (or share) your seed codes and (2) a bad actor cannot easily access your seed codes if your device is lost or stolen. I don't understand the community's desire for a security application to easily give away the secrets to anyone who possesses the device. Regardless of which side of

this argument you stand, let's assume you want to extract all of your seed codes from Authy and configure them with another provider.

First, make sure you have the Authy desktop app installed and configured on your Linux, Mac, or Windows computer, and it is completely closed. Next, restart the Authy desktop app in the following manner:

- Windows:** Right-click the Authy shortcut, in the Target field write “--remote-debugging-port=5858” at the end without quotes, click OK, and double-click the Authy shortcut
- Mac (Terminal):** open -a “Authy Desktop” --args --remote-debugging-port=5858
- Linux (Terminal):** authy --remote-debugging-port=5858

Open the URL of <http://localhost:5858> in any **Chromium-based** browser and conduct the following steps.

- Click the Twilio Authy link.
- In the Developer Tools top navigation bar to the right, click >> to expand the full list.
- Click “Application”.
- Under “Frames” expand “top”.
- Right-click “main.htm” and select “Open in containing folder”.

Next, paste the text from <https://unredactedmagazine.com/data/003.txt> beginning with “// QRious v4.0.2” and ending with “authy_backup.json’;” into the field at the bottom of “Console”. You may need to use cmd-v or ctrl-v to paste. This is a lot of text, which is generated from multiple programmers online. Each of the pieces of code includes attribution to the authors for further research, if desired. Continue to the next steps.

- Press enter on the keyboard to generate codes within the Console area.

- Click within this area and press ctrl-a (cmd-a) to select all content.
- Press ctrl-c (cmd-c) to copy the content.
- Paste it into any text editor of your choice.

The following image displays the Developer Tools area with the target section expanded. The image immediately after displays the generated tokens (redacted).



```
: Console
Filter Default levels ▾
1 Issue: 1
size: qr_size}}).toDataURL();
console_image(url, qr_size);
console.groupEnd();
return {name: i.name, secret:
secret, uri: totp_uri};
});
//console.save(data,
'authy_backup.json');

⚠️ Here's your Authy tokens: VM164:8
Twilio VM164:17
TOTP secret: VM164:17
TOTP URI: otpauth://tot VM164:17
VM164:6
```

My entries appeared similar to the following.

```
VM162:21 ProtonMail
VM162:21 TOTP secret: XXXXXXXX
LUP55L2B23VXXXXX2SX
VM162:21 TOTP URI: otpauth://
totp/ProtonMail%20
?secret=XXXXXXXXLUP55L2B
23VXXXXX2SX&digits=6&period=30
```

In this example, XXXXXXXX^{LUP55L2B23VXXXXX2SX} is the seed code for my Proton Mail account. I can now enter that into any other token-based 2FA account to generate identical temporary tokens as those present within Authy. You might consider the Standard Notes option which I present later in this issue.

There are a couple of lessons here. First, don't let YouTube internet bullies tell you how to live a private and secure life. Also, don't let me convince you to do anything. We all have different scenarios and are at unique places in our privacy journey. Make your own informed choices. Second, don't believe everything you hear when someone is bashing a privacy product on YouTube, which is ironically collecting data about every visitor and sharing it with Google. I still think Authy is a fine product, but I respect those who may want to move on to something else. You now have the tools to make the transition easy after you identify your own needs.



Image: Anh Nhat

USING A DE-GOOGLED PIXEL

By Zachary McIntosh

I have been daily-driving a Pixel 6 Pro running CalyxOS (May 2022 build) for a few weeks now. It's nice having a phone that isn't a 24/7 ankle bracelet for tech companies. I am pleasantly surprised at CalyxOS's usability and quite impressed at the level of control the OS gives you over almost all aspects of connectivity. Let me backtrack.

Following some key points of Michael Bazzell's exhaustive how-to book Extreme Privacy, I decided to try out a form of private phone usage. My threat model does not involve extreme hiding, but my primary motivation was to restrict information leakage to Big Data as much as possible.

A lot led up to this, starting years ago with deleting social media accounts: Facebook, Instagram, Snapchat, TikTok, and the other egregious

players in the "you are the product" space. I do use Twitter for announcing things, but purposefully and only on a compartmentalized desktop browser. I deleted all my iCloud backups and data, and stopped using Google services such as Wyze, Google Maps, Google Photos, and Google Docs. I stopped wearing my Apple Watch Series 6, and factory reset it. I even uninstalled the Amazon and eBay apps from my former iPhone 11 Pro Max, and only use these services on a desktop browser. I

canceled Audible, and factory reset my Amazon Kindles for giveaway. I then followed many recommendations from Cal Newport's book Digital Minimalism. I live a life of close to zero notifications on my mobile devices. I decided that I should determine when my attention goes to my phone, not the other way around. I mention all these previous actions because I believe a successful move to a private phone involves foundational habits and self-discipline that must already be in place.

Starting with the device, I followed Bazzell's protocol of buying the device with cash from Best Buy. I was originally set on getting a 128GB Pixel 6, but the looks of the Pixel 6 Pro just kept pulling at me. It also helped that the Pixel 6 was out of stock at the three Best Buys that I visited, with the second store telling me that another store in a different drivable city had only Pro's in stock. Decision by default.

I bought the Pixel 6 Pro (\$899 + tax) and a Mint Mobile trial SIM card (\$2), all with cash. The Best Buy employee mentioned saving \$20 by signing up with some sort of prepaid plan through a big 3 carrier, and I declined. They also asked if I had a number on record with Best Buy, and I said I did not. These stores really try to get a record of you. But cash is still king, so the transaction went fine.

Traveling to a parking lot by several food establishments, I powered on the device. It did the requisite Android setup steps, and welcomed me to my new Pixel 6 Pro. I skipped any registrations or network connecting. I followed the instructions on the CalyxOS website for Linux installation using my System76 Lemur Pro laptop. (I had loaded the page before I left home.) The CalyxOS site stepped me through how to enable Developer Options, enable USB debugging, and warned that the OEM Unlock Bootloader step might require internet to complete. I found this to be the case. I was parked near a McDonald's, so I did the following:

- To enable OEM Unlock Bootloader option in Android 12, I connected to McDonald's Free Wi-Fi.
- Connected long enough on this insecure public Wi-Fi to enable the unlock toggle.
- Android, sensing a network, started phoning home and downloading stuff in the background. It popped up a notification that it wanted me to "Finish setting up Pixel".

- I quickly disabled network after toggling OEM Unlock Bootloader and ignored any further setup prompts.
- I then ran the CalyxOS flash program on my laptop.

During first attempt, the device flasher script wanted to download platform tools v31 (even though I had already preemptively installed this). I connected my laptop to McDonald's Free Wi-Fi, but the download was failing me after several minutes waiting. I tethered my laptop to my personal phone hotspot to retrieve Android Platform Tools. As soon as it finished downloading I disconnected from Wi-Fi. The device flasher script proceeded, but then the phone reset to Google Android 12 factory settings. Trying again, the second time it started flashing when I allowed USB debugging on the Pixel.

After that, everything went smoothly. CalyxOS startup flashed a black circle that would grow and sweep on the screen. After a moment, I was greeted with the CalyxOS greeting screen and setup prompts. Nothing unusual to report there, I just finished setup and ended up with a stock looking Android home screen after several minutes. There were some cool non-Google replacement apps in place of what you'd normally see on Android.

I drove home and spent a bit of time without any SIM in the Pixel. I connected to Wi-Fi and downloaded Mullvad VPN. Then on VPN, I downloaded Aegis Authenticator, Yubico Authenticator, and Bitwarden password manager via F-Droid. I took this SIM-free downtime to get to know the OS and all its controls, all while having "Always-on VPN" and "Block connections without VPN" enabled. This phone was never going to talk to the internet without going through VPN.

I should probably mention why I chose CalyxOS over GrapheneOS. It all boiled down to overall Android app

compatibility offered by microG. I also prefer microG because it makes some attempts to simulate or replace aspects of Google Services for privacy; versus GrapheneOS which, despite having the capability to sandbox Google Services for security, still talks to Google Services. As I searched for Android app equivalents to my most-used iPhone apps, I used Plexus (<https://plexus.techlore.tech>) to see if an app would work in the microG environment. I also used Exodus (<https://reports.exodus-privacy.eu.org/en/>) to determine if an app had trackers or permissions that I would not want on my phone.

I am happy to report that the fingerprint reader on the Pixel 6 Pro works great. I was not expecting it to work, but it is much nicer to unlock the phone on the table instead of always having to lift it up with Apple's Face ID. After familiarizing with the quirks of CalyxOS, such as how the Datura Firewall works and how to set individual app permissions, toggle on and off the mic, camera, location, Bluetooth, etc., I felt like I could try connecting through a mobile carrier. I was excited to finally have 5G service.

I had to find a place with free Wi-Fi to register the SIM card. I brought an old factory-reset Pixel 3a to do the registration. Finding a Starbucks away from home but in the right area code, I hooked the 3a into the free Wi-Fi, created a new random Gmail account, and downloaded the Mint Mobile app. I activated the SIM card under an alias name. (Later, I would provide a [Privacy.com](#) virtual credit card number for subsequent billings.)

I later realized, after reading more about IMEIs and how cellular carriers work, that I made a huge mistake. I should have just activated the SIM in the new Pixel 6 Pro. Activating it with an old phone associated the old phone's known IMEI (along with all its previous subscribers) to the new Mint Mobile account. I got the 3a from eBay so who knows who the original owner

was. Not to say the carrier thinks all SIMs used with a particular IMEI are the same subscriber, but an association was made. Had I simply activated the SIM on my brand new Pixel, a never seen before IMEI would have a blank slate association to my new alias.

The weak point of anonymity and privacy while using cell phones is that the carrier will always track you: everywhere you've been, and all phone and SMS activity that you make on the phone, and anything else it can derive from your activity while you are on their network. Even though this is a MVNO with Mint Mobile, the parent carrier (T-Mobile) will track you along with everyone else around you, forever. Big Tech generally cannot get this data (I think), but law enforcement and government certainly can (and does) use this data. By having an alias account, despite the carrier data tracking everywhere you've ever been with this device, will not report it as tied to your real name. Oh well, let's move on.

I realized that I could turn off the SIM in CalyxOS via a network settings toggle. So, rather than going into airplane mode or Faraday bagging the phone when approaching home, I could simply disable the network that way. How trustworthy this software kill switch really is, I do not know. I have read that a carrier can still know your whereabouts even if the phone is not connected to network and/or when it is in airplane mode. But evading carrier tracking was not a priority in my threat model, so I figure I'd trust the software toggle. When home, I use Wi-Fi and VPN. When I leave home, I enable the SIM after driving a little while and use data that way. Later, I switched to Proton VPN for Android because it seemed to work more reliably on both Wi-Fi and mobile whereas I had some intermittent connectivity issues with Mullvad VPN on mobile data.

Transitioning off of the Apple ecosystem is difficult. I have been an iPhone user since 2007 when the first

iPhone was released by AT&T. A small exception is during the early 2010's when I dabbled with Android on an HTC EVO 3D that I got from Google I/O, but I quickly went back to Apple's products. One of the challenges to this phone being a daily driver and not just a "cool experiment" is having to face the fact that most of the United States uses iMessage. This includes my whole family and perhaps 99% of my friends.

My plan is to straddle the Apple world with my new privacy-enhanced CalyxOS phone. Would I have to carry around 2 phones all the time? That would be stupid. I discovered a client/server application called BlueBubbles. BlueBubbles server runs on a macOS machine that is logged into my Apple ID, and after some sketchy steps to disable Apple's System Integrity Protection, I am able to run a BlueBubbles server that has a ngrok endpoint. On the client side there is an Android app that connects to the server by scanning a QR code. From there, I can seamlessly send iMessages to friends and family as if I never left my iOS device. It is really quite clever. I suppose I am not truly cutting over solely to a new Pixel 6 Pro. But for the sake of staying compatible with the majority of my contacts, I keep my old iPhone and its phone number alive. Everyone is none the wiser. The old iPhone stays at home (and tracks me as always being home), but my everyday carry is my CalyxOS phone.

The only problem is I cannot FaceTime or call out from my old number, but more on that in a bit. So rather than trying to change the universe by asking everyone I hold dear to switch to Signal (or worse, Threema or Session), I can maintain conversations with them without any "green bubble" stigma. Over the past weeks, I noticed BlueBubbles server would sometimes stop responding. There is functionality to "restart server" remotely from the client, but this has worked only about half the time. It would be very annoying if it failed while I was far away from the home server.

Another problem that I faced having to straddle the Apple world for my family and friends was receiving phone calls. So, I followed Michael Bazzell's steps in setting up VoIP with Telnyx. I got a new number through Telnyx, and forwarded my old number to this. Also, I forwarded my known Google Voice number to this new VoIP number. I also followed Bazzell's recommendation for having text messages emailed to me with a small web hosted PHP script acting as a forwarding webhook.

Linphone for Android is a nice concept for phone calls. It's a portable number that can be picked up on my phone, desktop, or laptop. However, I can't say I had a good first impression of the mobile app. I installed Linphone (again by Bazzell's recommendation) via the Aurora Store. It would work while the app was in the foreground, and sometimes while the phone was not in use. But after about a week people were texting me that call attempts were dropping so I thought this might derail my whole effort with CalyxOS. On a whim, I uninstalled the Linphone app, and searched for it on F-Droid app store. This resolved the problem, because now I see that it has an attached background service that runs at all times to receive calls. Since the reinstall, Linphone has been receiving every call. I do miss call blocking features, which Telnyx does not provide. VoIP providers don't seem to have good ways of dealing with telemarketers and spammers.

One thing I cannot do is call out from my old iPhone number. If I have to make an outbound call, which is rare, I use Linphone and caller ID will show up as my VoIP number on the receiver's phone. I never want to use Mint Mobile's assigned phone number because I don't want the carrier having a record of my calls.

Likewise, I never send outbound SMS text. I use iMessage for most people and BlueBubbles integrates the SMS text messages and even handles short codes for me. To use my old and

widely known Google Voice number, I receive texts via email using Google Voice's forward SMS messages to email functionality. If I ever have to send out an SMS message, I open Chromium web browser and log in to Google Voice, then SMS text from there. I never intend to use my Mint Mobile number for SMS.

Now I have a phone that doesn't constantly transmit my personal data while it sleeps. I noticed some of CalyxOS's stock apps left much to be desired. I downloaded some Google apps via Aurora Store such as Google Camera and Photos, as well as Gboard, the superior swipe-text keyboard found on Android phones. What? Why would I do something like that if they are Big Brother apps? Well, thanks to CalyxOS, I turned off network access for each of these apps in Datura Firewall. They can try to phone home all they want; they won't get through. I get all of the functionality and none of the privacy loss.

For maps, I tested Google Maps but could not get it to work on CalyxOS. The app finds my position but the roads or satellite layers never appear. I use a navigation app called Magic Earth instead. It is a reasonable standalone app, where you download hundreds of megabytes of maps for offline use. It works well for exact addresses, but has difficulty finding establishments by name. The Organic Maps application that comes with the CalyxOS is okay, but I prefer the functionality of Magic Earth.

A lot is not mentioned here in terms of other apps, because app choices are highly personal. I just give a few examples. My phone is primarily used for 2FA, messaging, map navigation, taking photos, and setting up calendar events or reminders.

I also appreciate CalyxOS's (Android's) capability to set up separate profiles. My personal profile is named "IT Admin" to appear as a work managed profile, and I set up a

different profile with my name to serve as a dummy profile that I would switch to in case someone else needed to use or examine my phone. I just wish AOSP or CalyxOS developers would create a "duress passcode" functionality that could automatically switch to another profile based on the password you entered. That would be a cool feature.

Now for some of the bad news. After about a month of daily driving the Pixel 6 Pro with CalyxOS, I realized that some use cases aren't suitable for mission-critical things. My partner likes to call me after work, and I find that Linphone provides very poor call quality when forwarded from another number and funneled through VPN. Sometimes they can't hear me. Other times, especially on mobile data (even 5G), the call sounds like it's on the worst connection possible. Add to that the fact that Linphone rings, first asks for you to unlock your microphone in CalyxOS, then asks for you to unlock your phone to actually talk. Telnyx provides no useful help on how to improve call quality, and the application is locked behind too much security to answer quickly.

BlueBubbles has occasional issues. Sending/receiving image files, or worse, video files, is sometimes unreliable. There were times when a received photo would only partially show. I'd try "download again from server" via a menu option, and it would never complete the photo download. Initiating a new group chat has not worked for me. The group chat needs to be originated on the server host or on an iPhone in order to work properly.

Given this, the de-Googled life is good for most non-critical usage, and can cover perhaps 90% of my needs. But if I need reliable phone calls, I would need to use the Mint Mobile number to receive calls or to call out, or force my contacts to use a more reliable voice-over-data client like Signal.

Overall, I'd say a Pixel 6 Pro with CalyxOS is usable, private, and secure.

It feels like a lot of improvement is needed by the underlying replacement apps I have found. This is how stock Android felt compared to iOS in the past: it always seemed like a beta operating system just waiting to be improved. CalyxOS is much more usable than a Pinephone or Librem phone, but after these setbacks I think it needs more development time and polish to become a true alternative to existing mobile platforms. I continue to carry my Pixel 6 Pro around to reduce tracking and profiling. However, when I need to do something like travel out of state, I will probably tell contacts to just email or text me, and pull the iPhone out of a faraday bag if I ever need to make an important call or if the BlueBubbles server craps out while I'm far away. I'll just have that ready for things like Lyft, mobile banking, paying with NFC, reliable calls, and messaging, if needed.

I don't want to discourage the would-be user. We can't have everything. Having a private phone comes with some sacrifices that are manageable but not ideal. I do not regret moving to a Pixel 6 Pro with CalyxOS.

Having a de-Googled phone is a breath of fresh air. It is nice not having to worry what geofencing data, behavioral data, steps and flights of stairs traversed, accounts, device stats, app activity, telemetry, and other sneaky profiling data are being sent to a data miner or marketing firm. I think most people don't worry about mobile privacy because they themselves aren't conscious of what they do on their phones. But companies want to (and do) know. It's nice that I can stay connected, keep my data locked down, not change my whole family's habits, and still walk around with the benefits of a smartphone. ■



Image: ALEXANDRE LALLEMAND

THE RADIO RECEIVER: MONITORING NEIGHBORHOOD ACTIVITY

By Michael Bazzell

When I was a child, I was fascinated with police scanners. I felt mischievous as if I was doing something wrong while eavesdropping on my local emergency responders. One only needed to know the six-digit frequency of a government agency's radio system to listen to everything they did from a \$99 scanner. I could monitor activity in my neighborhood. If I saw the police lurking around, I turned on my scanner to get the details. Times were simple then. They are not now. Today, many agencies have transitioned to digital trunked systems which require special equipment and knowledge. We will tackle both, but let's first discuss the reasons why anyone would care.

I believe the best way to learn about the overall security of a neighborhood or city is to monitor police activity. One could request police reports from an area, but that can be misleading. Many calls for service do not warrant a report on file and the reports themselves only tell a specific part of the story. I prefer to monitor activity in real time. Before I purchase or rent a home on behalf of a client, I listen to police traffic in the area to determine the local criminal vibe. Even today, when I see a Sheriff vehicle near my home, I activate my scanner to know what is going on. This is how I keep aware of any problems within my community.

My preference for a desktop scanner is the Uniden BCD996P2 (<https://amzn.to/3mifGz4>).

This is a great all-purpose scanner with plenty of memory, a good speaker, and nice display. I believe it is one of the best "trunking" scanners in the \$350 price range. This is where things can get complicated. In the old days, a department had a few frequencies for analog traffic. As long as you were tuned to a frequency, you heard audio whenever someone pressed the button on the microphone. Today, most metropolitan areas have converted to digital systems which share numerous transmissions simultaneously on a single frequency. While this is a complex topic, let's break it down into a few digestible chunks.

First, you have a site. This may be a series of frequencies associated with

multiple towers at a specific location. In Los Angeles, the LAPD labels their sites as Northeast, South, West, etc. Each site possesses multiple frequencies.

The radio frequency may be similar to 850.000, but tuning your scanner to that frequency will not present any audio transmissions. That is simply the avenue for the digital transmission. This frequency might be the "control channel" or it could be a secondary frequency for digital audio. Either way, we hear nothing from this frequency. This is the most common complaint I hear about trunking. People buy a scanner, find their local police frequency from radioreference.com, but never hear anything.

Next, we have a talkgroup. This may be similar to 0001 for your police and 0002 for the fire department. This is not required to be programmed into your scanner, but it makes things much easier. It prevents you from memorizing numerous talkgroups.

The goal is to program your scanner with all of the local site frequencies and talkgroups desired. Programming the scanner exceeds the scope of this article, and is not always easy or straightforward. Spend some time with your scanner manual. Once you understand the programming of your model, radioreference.com contains all of the data you need. I program each site into its own bank, each frequency into that bank, and all available talkgroup IDs into the same bank. This way, when a transmission occurs, my scanner receives it. The scanner is monitoring the frequencies, stops when a transmission occurs, translates the digital transmission into audio, displays the talkgroup ID, and identifies the talkgroup by name. My scanner might light up with audio and I can immediately see that it is a detective unit for LAPD within the West site range. I could fill dozens of pages with other trunk scanning tactics, but that would be quite boring. Instead, I want to focus on some specific issues which seem to be common hurdles with trunk scanning.

The first is reception. Digital systems can be frustrating. With analog signals, weak transmissions are heard, but you may receive some static. With digital, it is all or nothing. If the digital transmission has a weak signal to your scanner, you will hear silence or broken audio. This is why antenna selection and placement is so important. I always recommend an external antenna whenever possible. If you live in NYC, you can probably get away with the free telescopic antenna attached to the scanner. If you are in the suburbs or a rural area, it will restrict your abilities. An external antenna will receive strong signals which your home's walls may block.

Antenna selection will make or break your scanning adventure. You should identify the frequencies which you want to monitor, and purchase an antenna appropriate for your needs. The best general purpose antenna is a Diamond discone. I like both the D3000 series (<https://amzn.to/3ax7h8o>) and the D130 options (<https://amzn.to/3MoePaL>). Note that the letters at the end of these refer to the types of connections. Make sure you buy an antenna with appropriate connections for your cabling, or purchase adapters to make them work. My cabling always contains a male "N" connector on each end, so I prefer antennas with a female "N" connector. I then purchase a female "N" to male "BNC" adapter to make it work with my scanner.

Discone antennas receive a broad range of frequencies. They work very well with lower, VHF, and UHF ranges. If your local agencies use frequencies within the 1xx.xxx through 4xx.xxx range, this will work very well, even for digital trunked systems. However, discones can be problematic if your local agencies use 800 MHz frequency ranges. I recently assisted a client with a move to a safe house. She wanted to monitor police in her area, but the site tower was several miles away. Her telescopic antenna was too weak and a discone did not pull in the 800 MHz frequencies she needed to monitor. I ordered a custom antenna from DPD

Production (dpdproductions.com) specifically designed for the frequency range she needed. That improved her reception drastically, and she can now monitor all appropriate talkgroups within that trunked system such as police, fire, and EMS.

Next is the cabling. If you have an outdoor antenna which requires more than 15 feet of cabling, and you need to monitor higher frequencies in the 700 or 800 MHz range, I strongly suggest LMR-400 cabling (<https://amzn.to/3MC8Dwd>). This is more expensive, but much better shielded to prevent escape of higher frequencies. If you have sites which are hard to lock into, better cabling alone may be all you need.

Is this all worth it? Only you can decide. I often leave my scanner on throughout the day. It has alerted me to nearby wildfires which would be "breaking news" hours later, an elderly neighbor alone with a medical emergency and a 30 minute EMS response time, and early notice that road crews would be blocking the primary exit from my neighborhood for a water line break.

Next, I always program common frequencies associated with personal radios. These include Family Radio Service (FRS), Multi-Use Radio Service (MURS), General Mobile Radio Service (GMRS), and Citizen Band (CB). These will allow you to monitor personal radios being used in your neighborhood. If you go to the local department store and purchase a pair of two-way family radios, you will be transmitting on a public radio frequency that can be monitored with a scanner. The Family Radio Service (FRS) is a radio system authorized for use without a license since 1996. There are 14 FRS channels available which operate on 14 specific frequencies. The following identifies these frequencies.

- Channel 01 - 462.5625
- Channel 02 - 462.5875
- Channel 03 - 462.6125
- Channel 04 - 462.6375

- Channel 05 - 462.6625
- Channel 06 - 462.6875
- Channel 07 - 462.7125
- Channel 08 - 467.5625
- Channel 09 - 467.5875
- Channel 10 - 467.6125
- Channel 11 - 467.6375
- Channel 12 - 467.6625
- Channel 13 - 467.6875
- Channel 14 - 467.7125

The most common use for these radio frequencies is by families on vacation or at large events. They allow parents to keep in contact with their children. Criminals have found uses for them as well. Subjects often referred to as "spotters" use them to notify drug dealers when police are approaching a specific area. Additionally, illegal business operations such as gambling rooms and prostitution houses will use them to communicate cheaply and "anonymously". While this may afford the users some privacy protection against personal identification, the transmissions are completely public. Since the transmissions can travel several miles, the audio can be intercepted safely and without detection. Programming and monitoring these frequencies in known criminal areas may provide raw intelligence about your neighborhood. These are not the only frequencies to consider.

The Multi-Use Radio Service (MURS) is an unlicensed two-way radio service that was established in 2000. The radios are capable of a range of ten miles when using decent antennas. The following table identifies the MURS frequencies.

- 151.820
- 151.940
- 154.600
- 151.880
- 154.570

Additional public frequencies, known as General Mobile Radio Service (GMRS) frequencies require a license to legally transmit audio. Most

people ignore this requirement and it is seldom enforced. The radios that transmit on these frequencies can use up to 50 watts, allowing the signal to travel farther. All MURS and GMRS frequencies should be programmed and monitored in the same way as FRS frequencies. The following list identifies the GMRS frequencies.

- Channel 01 - 462.550
- Channel 02 - 462.575
- Channel 03 - 462.600
- Channel 04 - 462.625
- Channel 05 - 462.650
- Channel 06 - 462.675
- Channel 07 - 462.700
- Channel 08 - 462.725
- Channel 09 - 467.550
- Channel 10 - 467.575
- Channel 11 - 467.600
- Channel 12 - 467.625
- Channel 13 - 467.650
- Channel 14 - 467.675
- Channel 15 - 467.700
- Channel 16 - 467.725

Many people associate Citizen Band (CB) radios with truck drivers. This is often appropriate, but truckers are not the only people that transmit on such frequencies. Since CB is low power, the receiver must be within a few miles of the transmitter. There are 40 channels available in this band. Communication on these channels may include traffic issues, witnesses to major accidents, reports of reckless drivers, and the occasional sermon. Many state patrol vehicles include a CB radio for receiving and transmitting. The following list identifies these channels and frequencies.

- Channel 01 - 26.965
- Channel 02 - 26.975
- Channel 03 - 26.985
- Channel 04 - 27.005
- Channel 05 - 27.015
- Channel 06 - 27.025

- Channel 07 - 27.035
- Channel 08 - 27.055
- Channel 09 - 27.655
- Channel 10 - 27.755
- Channel 11 - 27.085
- Channel 12 - 27.105
- Channel 13 - 27.115
- Channel 14 - 27.125
- Channel 15 - 27.135
- Channel 16 - 27.155
- Channel 17 - 27.165
- Channel 18 - 27.175
- Channel 19 - 27.185
- Channel 20 - 27.205
- Channel 21 - 27.215
- Channel 22 - 27.225
- Channel 23 - 27.255
- Channel 24 - 27.235
- Channel 25 - 27.245
- Channel 26 - 27.265
- Channel 27 - 27.275
- Channel 28 - 27.285
- Channel 29 - 27.295
- Channel 30 - 27.305
- Channel 31 - 27.315
- Channel 32 - 27.325
- Channel 33 - 27.335
- Channel 34 - 27.345
- Channel 35 - 27.355
- Channel 36 - 27.365
- Channel 37 - 27.375
- Channel 38 - 27.385
- Channel 39 - 27.395
- Channel 40 - 27.405

I hope this article serves as an introductory to neighborhood radio scanning. As with the previous article, these first few submissions exist to establish the groundwork for future discussions. Once you have an understanding of the ways to receive various radio signals, we will have unlimited areas to explore. See you next month. ■



Image: Jeremy Bezanger

MITIGATING RISK OF ONLINE SERVICE FAILURE

By Oyzark

In the privacy community, we experience a higher risk than most people of having the online services we rely on suddenly become unavailable to us. There are several reasons for this: we tend to use smaller, newer and lesser established platforms such as Proton Mail instead of the behemoths such as Gmail used by the masses; government actions can force sudden closing of privacy-oriented services (think LavaBit); and in pursuing privacy we tend to exhibit non-typical behavior patterns that can trigger alerts on our accounts.

There have been several reminders of this recently, including the sudden closure of the CTemplar email service, and Michael's experience with Telnyx phone service as described in a recent podcast. Personally, I just lost my SudoMax 9-phone number account for a couple of days due to some kind of system glitch. Fortunately their excellent support got it back up and running quickly, but it serves as a reminder that all systems are vulnerable.

In this article, I describe steps I have taken to mitigate risks to some of my most important online services - email, contacts, calendar, notes, messaging

and phone. These are somewhat specific to my situation, but I think the steps are generic enough they could be implemented by others.

Email, Contacts and Calendars

A Proton Mail Professional account (protonmail.com) is the hub for all my personal email activity, as well as personal contacts and calendar. If the account disappeared one day without mitigation, it would have a pretty significant negative impact, including losing friends and family contact information, losing precious old emails, missing new emails coming in, and having no idea what events I am

supposed to be at in the future. This is all important enough that I need to keep a backup, but also get back on my feet again within hours if my Proton Mail account were to disappear. Here I describe a validated approach to using Tutanota as my alternate provider in case of loss of Proton Mail. The first steps are to ensure that all my important information is backed up somewhere regularly outside of Proton Mail; then comes ensuring I have an alternate, tested system that would get me back on my feet using email, contacts and calendar in hours, even with Proton Mail gone. Here are the steps I took:

1. Switch to a custom domain in Proton Mail. Rather than using zzz@protonmail.com, I have Proton Mail host my own domain, so my email is more like zzz@mydomain.com. This has several advantages including avoiding problems caused by services not liking Proton Mail domains, and meaning I can quickly switch my email provider to another one without changing my address. You do need a paid Proton Mail account for this, and you do need to buy a domain from a registrar like GoDaddy or Namecheap. Proton Mail has instructions on setting up a custom domain in Proton Mail (<https://protonmail.com/support/knowledge-base/set-up-a-custom-domain/>).

2. Create an offline backup process for existing email. The easiest way to do this is to run Proton Bridge (only available on paid accounts) and use an offline email program, such as Thunderbird, to archive mail. Thunderbird will ingest email continuously (when open) and save it locally in your profile folder. This profile folder can be copied to backup storage. To find the profile folder location in Thunderbird go to *Help -> More Troubleshooting Information* and click *About Profiles*. You will then see the root directory that is used for your current profile. The beauty of this approach is that Thunderbird has excellent offline search capabilities, so you get some helpful functionality beyond backup; and if I did lose my Proton Mail I probably would just leave old email in Thunderbird and access it there. An alternative is to use the

Proton Mail Import-Export app (<https://protonmail.com/import-export>, also only available for paid accounts) that saves an MBOX file to your local storage. This is a manual process, but you can create a full backup just once, then create incremental backups of your mail by setting date ranges in the backup dialog. Most email software and providers allow import of MBOX files, so you are relatively safe here. I decided to do both Thunderbird and MBOX files. Do be aware that this can really use up some disk space, and exporting to the MBOX file can take several hours if you have a lot of mail.

3. Create an offline backup process for contacts. This is a simple, but manual process. Click on the contacts icon in Proton Mail, then *Settings*, then *Export Contacts*. Your contacts will be decrypted and saved in the standard VCard (.vcf) format.

4. Create an offline backup process for calendar. Similarly, in calendar go to *Settings* then *Calendars*. Beside each calendar is a drop-down button with the option *Export ICS*. This will save a standard .ics calendar file with all your appointments. An alternative is to create a calendar link under *Share outside Proton*. This will give you a URL you can use to ingest your calendar into another calendar app, such as the one in Thunderbird. Make sure this app keeps an offline copy of your calendar entries. The link also gives you a quick way to save an ICS file without being in Proton Calendar - just paste the URL into a browser and it will prompt you to save the file.

5. Set up Tutanota as backup provider for email, contacts and calendar. It was very straightforward to make a new Tutanota account. I could have gotten away with just keeping a free account and upgrading later if needed, but I wanted to test the domain hosting capability, and so decided to purchase the "Business Premium" account (€24/year) which allows multiple custom domains. [Privacy.com](https://privacy.com) payment was accepted.

6. Test it out! This is the most important step. If you don't fully test

your mitigation plan, you can be sure there will be a fatal flaw when you really need it. I went through the whole process of hosting a test domain in Tutanota, then ingesting my contacts and calendar entries. The testing brought up two issues. First, Tutanota does not currently have the ability to import emails (it is on the roadmap). This is not a showstopper, as I am okay leaving my old mail in Thunderbird and accessing it there when needed. I did note though that Tutanota does not have a batch export capability either which could be an issue down the road (also on roadmap, although you can export individual emails). Second, Tutanota gave an error when trying to import the contacts exported from Proton Mail. After some exploration, this seems to be because Proton Mail is exporting vCard version 4.0, whereas Tutanota seems to be expecting an earlier version. I was able to fix this with the sed tool on Linux, to replace the version number in the file (replace the filenames with yours): `sed 's/VERSION:4.0/VERSION:3.0/' 'protonContacts-X.vcf' > protonContacts-X-Tutanota.vcf` Other than that, importing the contacts and calendar into Tutanota was straightforward. Adding a custom domain is explained briefly at <https://tutanota.com/faq#custom-domain>. Clicking on *Global settings -> Custom email domains -> Show* then clicking the add button will get you started. I did this, and it worked well.

Notes

I use Standard Notes (standardnotes.com) for organizing a lot of fairly random but important textual information. Standard Notes has excellent backup options, including options to save encrypted or unencrypted backups directly to disk, or have them sent regularly to you by email. Encrypted backups can be decrypted back into text or an unencrypted Standard Notes import file using an offline browser-based decryption script available at <https://github.com/standardnotes/decrypt>. Of course just having lots of plain text isn't super useful, and I need to be able to quickly restore a Standard-Notes-Like interface should Standard

Notes become non-functional. For this I chose Joplin (joplinapp.org), an open-source note app. It's not quite as flexible as Standard Notes, but would do the trick in a pinch.

Getting notes from a Standard Notes backup into Joplin takes a couple of steps, made much easier with a Python script available at <https://github.com/tanrax/standard-notes-to-evernote-or-joplin>. This script transforms an unencrypted Standard Notes backup file into an Evernote ENEX file called *notes.enex*, that can be imported directly into Joplin. If, like me, you like to store your Standard Notes backups as encrypted, you have to do a bit of wrangling to get the output of the decryption script into a format that works for the Evernote script. Specifically, the Evernote script expects the text to be in a file called "Standard Notes Backup and Import File.txt" that is in a compressed (ZIP) container. So the steps you need to do starting with an encrypted Standard Notes Backup file are as follows (for Linux, you can do similar on other operating systems):

1. Decrypt the backup file using the web-based decryption script, and choose "download as decrypted import file". This will create a file called *decrypted-sn-data.txt*
2. Rename this file: `mv decrypted-sn-data.txt 'Standard Notes Backup and Import File.txt'
3. Compress this renamed file: `zip sn.zip ./Standard\ Notes\ Backup\ and\ Import\ File.txt`
4. Run the conversion to ENEX script (you will need Python installed): `python3 standard-notes-to-enex.py ./sn.zip`
5. Import the notes.enex file into Joplin (File -> Import -> ENEX (as Markdown))

Messaging and phone

Messaging is quite straightforward, as like most people I only use text messaging for ephemeral communications that I don't need to

persist for long. Thus backups and archiving are not really necessary, but redundancy of service is. I currently use Signal and Wire as my primary messengers, and have a few others like Element/Matrix and Session for experimental or backup use. While their usage is differentiated, they do all serve as a backup to each other. So for really important contacts, I try to connect with them on at least two messenger services, so if one goes away, then we can use the other. I also where possible ensure I have an email address for anyone I am contacting on a messenger, so if, for instance, my account gets disabled on one of the messaging services, I can easily correspond by email once I have a new account set up. Signal does, of course, have a critical dependency on a phone number, which I think is a vulnerability as the number itself is under the control of a third party (see below). So I am starting to favor messengers that don't require a phone number linkage.

Unlike messaging, phone numbers are a real headache for risk mitigation. The problem stems from two realities that are at odds with each other. The first reality is that *keeping* a phone number, if you are a privacy enthusiast, is actually quite a lot of work and to a large degree out of your control. Your phone number is owned not by you, but by whatever VoIP, landline, or cellular provider you lease it from. If that provider goes out of business, has a technical failure, or simply decides they don't want to do business with you anymore, you either lose the number, or if you are lucky you manage to port it to another provider through an unreliable, clunky process called porting. The second competing reality is the *legacy social expectation* that you will have a "phone number" and that this number will persist for years or even decades. So you have to act as if your phone number is virtually part of your identity, yet you have little control over its persistence.

It's 2022, why do we need to loan a 10-digit number in order to be able to communicate with people? The truth

is we don't. We are so conditioned to our "contact information" being name, email address and phone number, that we don't think about how silly it really is given the many communications options available to us. So the shift I am making is as much a pushback of social expectations as a technical mitigation.

Here's my strategy. I own a long-standing Google Voice number. This number forwards to whatever VoIP number I am using currently, along with my office phone. SMS messages are sent to my email. Of course Google could pull this number at any time, but given all the options available to me I think it is the most *stable* number I have, even if it's not the most *private*. When someone asks for my phone number, I respond with something like: "I'm in the midst of switching providers right now (always true!) so the best way to contact me is by (email/messenger). If you really need to use the phone you can try calling my Google Voice number XXX and I should get the message". In this way, I am minimizing the expectation that using a phone number for me will be effective, while giving a modestly reliable option if they absolutely have to use a phone. This means I have a somewhat stable personal phone number when needed, freeing me to use VoIP services for more "disposable" numbers, such as for forwarding from my main number or for temporary aliases. This is not perfect, and I'm constantly adjusting my strategy here.

Summary

What I most want you to get out of this article is a sense of the importance of planning in detail an alternate strategy for the online services that are important to you. It's only a matter of time until you experience a service failure, and if you plan for and mitigate such a failure you can minimize the disruption to your life. I can now sleep well knowing I have a tried and tested mitigation strategy for my most important stuff. ■

A FRESH LOOK AT STANDARD NOTES

By Michael Bazzell

I have been using Standard Notes for many years. I started with the sole free option when it first arrived, experimented with paid features introduced later, and settled for the free version over the past few years. It is a staple in my daily computing. Standard Notes is an application which allows strictly end-to-end encrypted notes which can be synchronized through their servers. Any notes maintained on my computer are immediately synchronized to the same app on my mobile device. No one outside of my applications can see the content, even employees of Standard Notes.

The free version has suited me well. I always preferred that it was limited to plaintext content. I could create unlimited notes, and plaintext copy/paste functions ensured that there was no formatting included within my content. I have maintained an outline and notes for the last ten books I have written, all of the notes about pending podcasts, and every outline for this very magazine within Standard Notes. It has been the way I safely organize my notes for many years. I can't imagine digital life without it now. Therefore, I thought I should take a new look at the paid features.

Standard Notes offers three tiers. The free tier is limited to plaintext, the Productivity tier (\$59 annually) offers enhanced options, and the Professional tier (\$99 annually) offers larger file

support and 100GB of storage. I went with the highest tier so that I could fully test the features.

The biggest change with any paid tier is the availability of new formatting for notes. You can choose rich text options and utilize bold, italic, new fonts, and everything else you would expect with a note taking application. I played with this for a while, but I missed my plaintext notes and eventually reverted to that setting. Having an option for markdown text was nice, but I never used it. I keep things simple, and really didn't care much to complicate my notes with visual enhancements.

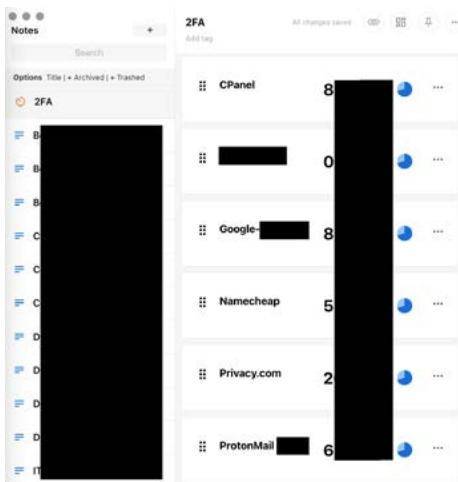
Then I noticed the spreadsheet feature. This was a game changer. I immediately created a new note, switched the format to spreadsheet, and possessed a simple Excel replacement. I now use this hourly. It is great for documenting things I wish to later alphabetize or spread out visually. It solves the problems with tab-based plaintext files and allows me to navigate through data easily. I find this option better for my podcast notes than text now. It allows me to make notes within any area of the file, even on the fly. This was worth the price alone. I rarely open any official spreadsheet application now. The image below displays my recent activity while DXing on my short wave radio and picking up new international stations. Clicking the "Data" tab and then the down arrow allowed me to alphabetize the frequency column for quick visual access. My note today has

hundreds of frequencies, call signs, and locations all neatly organized. No more OpenOffice for this simple task.

	A	B	C	D
23	Freq	Station	Location	
24	4840	WWCR	Tennessee	
25	5085	WTWW	Tennessee	
26	4765		Cuba	
27	5025		Cuba	
28	5800			
29	5850			
30	5935	WWCR		
31	5950			
32	6030			
33	6105			
34	6165			
35	6185			
36	7335		China	
37	7505	WRNO		

Next was the TokenVault authentication option. It would be easy to say this is just another simple 2FA token storage option, but that would not do it justice. Yes, its only purpose is to store 2FA seed codes and present a temporary token which changes every thirty seconds. However, that is not the headline here. It is a completely open source, end-to-end encrypted, cross-platform, non-password manager based, software token generator which

is synchronized through your encrypted Standard Notes account. I do not know of any other option which checks all of these boxes. If you read my previous article on exporting your Authy codes, you could import them into Standard Notes for a full 2FA solution. **This is the understatement of this issue in my opinion.** I now use Standard Notes for all of my software-based 2FA tokens. For further protection, I password protect this note. The following image displays a heavily redacted partial view of my 2FA note. The ability to sort these alphabetically is a simple but huge piece. My Authy Android application forced me to drag-and-drop all entries in the order desired.



I had hundreds of seed codes which I had exported from Authy, so I spent some time identifying a way to import them into Standard Notes. Actually, I am sure I spent more time on this than would have been required to enter them manually. In the end, I took all my exported Authy codes, which appeared similar to the following.

VM164:17 CPanel

VM164:17 TOTP secret:
xxxxxxxxxx

VM164:17 TOTP URI: otpauth://totp/CPanel?secret=xxxxxxxxxx&digits=6&period=30

VM164:6 +

VM164:17 Twitter

VM164:17 TOTP secret:
xxxxxxxxxx

```
VM164:17 TOTP URI: otpauth://totp/Twitter%20IT?secret=xxxxxxxxxx &digits=6&period=30
VM164:6 +
```

I then used various sed commands to make them each look similar to the following.

```
{
  "service": "CPanel",
  "account": "",
  "secret": "xxxxxxxxxx",
  "notes": ""
},
{
  "service": "Twitter",
  "account": "",
  "secret": "xxxxxxxxxx",
  "notes": ""
},
```

I then exported all notes, unzipped the archive, modified the archived 2FA note with my new entries, re-zipped the archive, and imported it back into Standard Notes. The result was all tokens present and waiting for use. Again, it would have taken me less time to manually enter each seed code, but I wanted to see if this would work. Out of caution, I will not share my sed commands. If things were to change over at Standard Notes, my steps could wipe out valuable data. I have spoken with the CEO of Standard Notes about a seed code import utility, and he seemed interested. If you purchase a paid tier, send them a reminder of the usefulness of this potential feature. You can currently export your 2FA entries into a single JSON file and securely store them for future use.

By default, Standard Notes is secure and private. All notes are completely encrypted. However, we should always take advantage of any additional features available to us. First (and obvious), choose a strong password for your account. Enable

two-factor authentication in the Settings > Security area, and consider email notifications whenever someone signs into your account. Make sure you have a way to access your 2FA for the application. If you only use Standard Notes for your 2FA, then you need access to a valid session in order to log into another. Since I have the application synchronized to three devices, I am not concerned about this. However, I always have my exported seed codes if I need them. I could use KeePassXC to generate a new token if desperate. I disable the session logging for more privacy. If desired, you could add a passcode lock which would prevent anyone with physical access to your devices from seeing any content without the code. You can also protect individual notes for anything sensitive. Make sure that your note backups are encrypted at Settings > Backups. This prevents anyone from viewing your text-based backups on your local drive. This should be the default setting.

The premium tier offers support for large files and immense storage, but I have yet to use any of it. In fact, I have yet to rely on any features in the top tier. I would encourage others to explore the middle paid tier and see if it would meet all of your needs. Navigate to <https://standardnotes.com/plans> and identify the plan best for you.

I have always liked Standard Notes. I was a freeloader for years. I never thought the “fancy” paid plans were needed for my usage. I was wrong. I can never go back. I am slowly identifying additional features of the paid tiers which can be incorporated into my daily use. The to-do task editor provides check boxes which I can tick to make me feel productive. Regardless of your tier, even if using the free version, make sure you are backing up your notes for offline use on another machine if ever necessary. I have never needed them, but feel prepared if I do.

Standard Notes is not a sponsor of this magazine or the podcast, and they did not request a review or offer any input for this article.



Image: Dan Dimmock

THE OSINT CORNER ↗ LEARNING THE LINUX COMMAND LINE

By Jason Edison

Jason instructs live and online open source intelligence courses for IntelTechniques in addition to working as a cyber-crime detective for a large U.S. police department. Each issue will feature an OSINT tactic from the IntelTechniques online training.

One of the most daunting aspects of moving from beginner to advanced OSINT work is incorporating Linux scripts and virtual machines into your workflow. Use of Linux and the command line is a steep learning curve, especially for those of us who are primarily Windows users. The transition to Linux from Mac is typically smoother because both are Unix based systems, but even Mac users may find some of the following tips and resources valuable. Keep in mind that I am primarily a Windows user and this advice is based on my own experiences transitioning to the Linux command line. There is no singular correct approach to learning a new skill and your own chosen path may vary.

Tip #1 Terminology: For the purposes of this article, we are going

to refer to the “command line.” You will see this also referred to as terminal, shell, and console, although technically they are not the same thing. We are going to keep things simple and use “command line” in the context of entering text-based commands into the Linux terminal. For a more detailed comparison of these terms please see <https://www.geeksforgeeks.org/difference-between-terminal-console-shell-and-command-line/>.

Tip #2 Time: Learning complex skills takes time and it is important to control our expectations. Repetition is key in moving our skills to a place where they are second nature, and that is going to require investing time putting fingers on keyboard. We need to accept that proficiency takes time to build. As we often say, it is a marathon, not a sprint.

Tip #3 Immersion: I often am asked what is the fastest way to get comfortable with the command line. That answer is simple: force yourself to use it for everything. This can be very, very slow and frustrating at first but just like learning a foreign language, the fastest way to learn Linux is to immerse yourself and start building repetition. Resist the temptation to move back to your Windows or Mac workstation, and try to avoid using the graphical interface on your Linux machine. You don't need to go out and buy a new computer to dive into Linux. A good starting point is to build out an Ubuntu virtual machine and work through customizing it for OSINT work as detailed in Michael's books. While you are manually customizing your virtual machine with our recommended commands, it is good to have a browser

open on another screen so that you can research any commands which you do not understand as you go.

Tip #4 Practicing Linux on Non-Linux Workstations: The preferred method for practicing Linux on your non-Linux workstation is to create a Linux virtual machine using a hypervisor such as VirtualBox or VMWare. This will give you an experience that is very close to working on an actual Linux workstation but without having to buy new hardware. If you do not have access to the steps in our books and online training, a basic overview of setting up a virtual machine can be found at <https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#1-overview>. Macs are also Unix-based systems so many of the commands and scripts carry over, but there are differences. I recommend learning in a true Linux environment rather than in the terminal on your Mac computer. It is also possible to run a limited Linux environment on our Windows workstation using a feature called Windows Subsystem for Linux.

Unfortunately, enabling WSL2 on a Windows workstation tends to break VirtualBox. Currently I recommend avoiding WSL2 unless you are an advanced user who feels comfortable troubleshooting any conflicts. Most people will be better off avoiding WSL2 and using a full virtual machine.

Tip #5 Training Resources: Although immersion is the fastest way to learn, it may not be an option for everyone. If you are unable to go all in, but want to start working on your Linux skills, below are some additional zero-cost training resources.

Linux Basics Lessons:

- <https://ubuntu.com/tutorials/command-line-for-beginners/>
- <https://linuxjourney.com/lesson/the-shell/>
- <https://www.youtube.com/c/LearnLinuxTV/>
- <https://www.makeuseof.com/tag/an-a-z-of-linux-40-essential-commands-you-should-know/>

• <https://linuxcommandlibrary.com/basics/>

• <https://explainshell.com/>

Online Terminal Emulators allow us to practice command line basics from any browser:

• <https://linuxsurvival.com/>

• <https://cmdchallenge.com/>

Tip #6 Patience and tenacity:

This circles back to where we started in managing our expectations and dedicating the time to build up these new skills. Frustration is the most common roadblock to gaining proficiency with Linux and the command line. I see many people give up early on because of the steep learning curve, so please pace yourself and set reasonable goals. Know when to step away and return later with fresh eyes, renewed patience, and a rested mind. ■



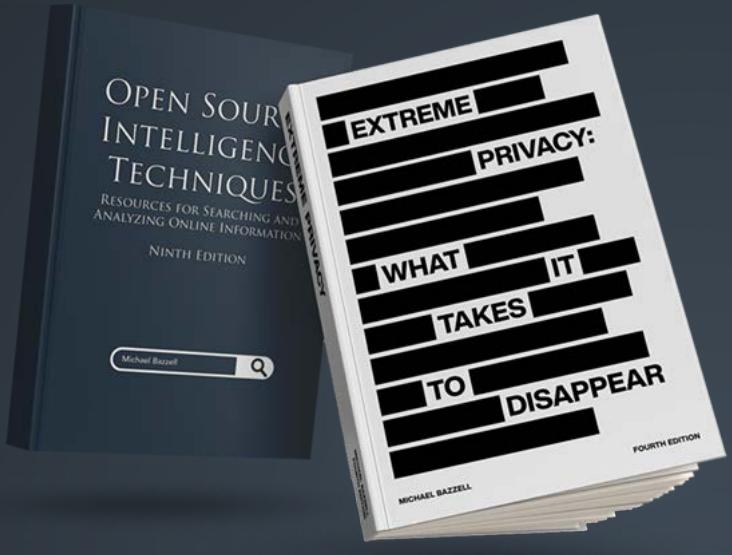
INDUSTRY LEADING TECHNOLOGY AND A 24X7 SOC WORKING FOR YOU

Cyber threats are evolving rapidly. SMBs and Enterprise businesses are looking to their Managed Service Providers to provide them with cybersecurity solutions. Our managed SOC is highly-skilled in the constantly evolving threat landscape and will provide absolute security for you and your clients.



FORTIFY24X7.COM | (800) 989-2647 | INFO@FORTIFY24X7.COM

New 2022 Privacy & OSINT Books



- ✓ Hardcover & Paperback
- ✓ New & Updated Content
- ✓ 500+ Pages Each @ 8.5 x 11
- ✓ Our Full Playbooks
- ✓ Supports This Free Magazine

Order at IntelTechniques.com

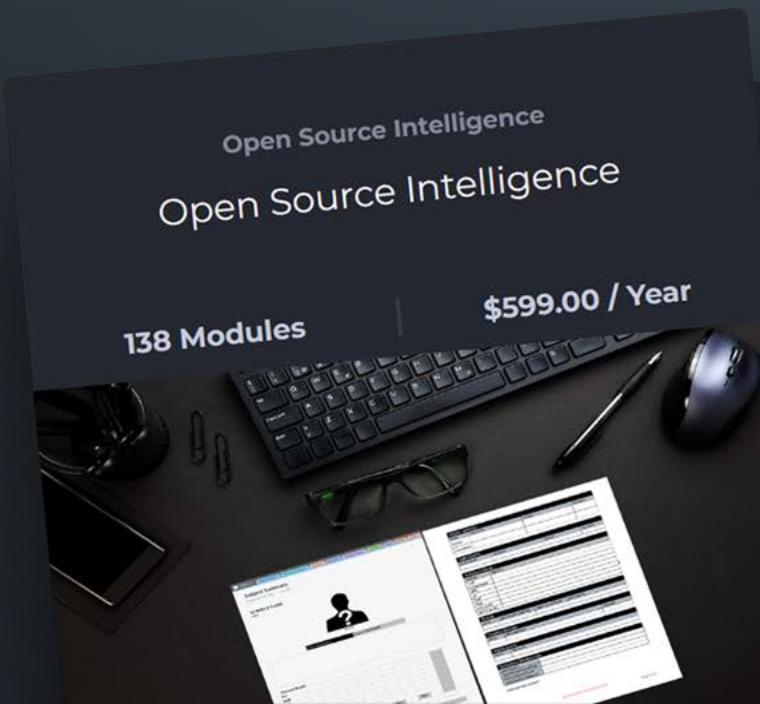
OSINT & Privacy Video Training

90+ Hours of Video Training | Optional OSIP Certification

Register at IntelTechniques.net

Open Source Intelligence
Video Training + Certification

\$899.00





HP DEV ONE WITH POP!_OS

By Michael Bazzell

I typically avoid computer reviews. Everyone is unique and has their own needs. My favorite computers are those you can customize for your own usage, which may negate the need for a review in the first place. If you have been following along with my full-time transition to Linux Pop!_OS, you know that I rely on a System76 Thelio desktop for my daily computing. I chose a desktop because I work with 20TB of breach/leak/logger data daily and the desktop allows me to “turn off” better at the end of my work day. I chose a System76 desktop because it natively supports Pop!_OS and was designed

with Linux users in mind. You can read much more about that in the previous issue.

After my article about Linux basics (issue 002) and the corresponding podcast (264-Back to Basics-Linux I), many readers and listeners notified me that four of the six System76 laptops were now sold out. Several people were asking me for a recommendation of a solid laptop which supported Pop!_OS out of the box and was affordable. That can be a tall order.

I began my search and immediately noticed a trend with the popular laptop manufacturers. Lenovo offers two Linux-ready laptops, but both have

issues. The processors are old, the software is outdated (Ubuntu 20.04), and the prices are high (> \$2,000). Dell does a bit better with multiple options, but all still come with Ubuntu 20.04. One could obviously install a newer version of Linux, but I was looking for something which would just work without any driver issues. The specs on the Dell options were much better than Lenovo, but the prices still seemed high. Then I found something I did not expect to see. HP just launched a new laptop in partnership with System76 which includes the latest Pop!_OS installed. I did a double take.

Since this was a brand-new machine, I knew I could not go to the local Best

Buy to check it out. I contacted their media relationship office and requested a demo unit. It arrived today, and I am pleasantly surprised with it. Let's dive in.

This is a 14" laptop with an AMD Ryzen 7 processor at 4.4 GHz max (8 cores / 16 threads). 16GB of RAM is installed within two slots, and is replaceable. A 1TB NVMe drive has a max speed of 3,000 MB/s. The graphics are powered by integrated AMD. One should note that this is a 14" laptop with a max resolution of 1920 x 1080. If you are looking for Retina-like high resolution on a huge screen, this is not for you. The display may be a weak point, but I run my desktop at 1080, so it was fine with me. I always prefer integrated graphics, especially within laptops, as they are more energy efficient, and therefore typically offer longer battery life.

Speaking of battery, the unit claims 12-hour usage between charges. My initial setup of applications and all configuration lasted nine hours, but that was under heavy load. The 12-hour average seems legitimate.

Both USB-A and USB-C ports are on the sides along with traditional HDMI. So far, this thing is checking all the boxes. However, the big deal is the price: \$1,099. I know one should never choose a laptop on price alone. However, this is a huge benefit. I am always looking for a slam-dunk when it comes to a pre-configured laptop I can recommend to clients. Let's compare to Lenovo and Dell. A Dell 13" XPS with similar specs is \$1,750. However, this includes only 8GB of RAM and outdated software. A similar ThinkPad was \$1,900 with Ubuntu 20.04. \$1,000 is a very attractive price for a powerful Linux machine, but let's discuss the unit itself.

The HP Dev One is targeted toward developers. They have recognized that there is a growing community

of programmers who need reliable hardware running on Linux which does not need to buy the latest gaming laptop with flames on the side. However, I think this limited scope is excluding a larger crowd. This machine has specs which would work for any Linux transition.

The unit feels solid. One complaint I hear about niche privacy laptops is that they can have a very light "plasticky" feel. This does not have that. The size is great and the machine feels very slim and portable. The keyboard is firm and immediately usable. The screen is bright and clear. The embedded physical webcam cover is a nice touch. We are all spoiled today.

As I have stated in previous articles and podcasts, I rely on a laptop while on the road, which can be often. As I am travelling this week, I am solely using the HP Dev One. I immediately observed two things. First, now that I am using the new Thelio desktop, my four-year-old laptop feels dated and slow. Second, the HP Dev One laptop has almost the same overall experience as the faster desktop. Let's talk specs.

The processor and RAM are plenty fast. I don't think many readers would notice the difference between this and a faster processor, especially using Linux. Everything seems snappy and immediate. The 16GB of RAM is sufficient for most people. I was able to launch virtual machines fine, but loading three at once understandably caused some lag. I may swap out the two 8GB chips for two 16GB for a total of 32. Replaceable parts are nice.

Booting the machine presents the latest version of Pop!_OS. The only difference I saw was the option to send analytics. They made it very clear and easily offered the option to avoid this. They even allow you to see what a report would look like and the option to remove your own data from their servers at any time. I was impressed

at the transparency and control. I understand the need to learn about any errors or crashes in order to make the product better, but I always choose to decline sending of any data due to my privacy paranoia.

Once booted, it was the typical Pop!_OS experience. You receive all updates directly from System76 and everything on the machine just works. It doesn't feel like a computer which was repurposed for Linux. It feels like a native Linux computer. It also feels very obvious that Pop!_OS recognizes this specific computer and hardware, which eliminates many headaches and presents a turn-key experience.

HP is also offering the System76 Launch keyboard and their own programmable mouse. I can see how both of these would be beneficial to developers, but I am not sure many readers would take advantage of reprogrammable keyboard and mouse. However, Pop!_OS includes a utility created by System76 to make the modifications easy. It is on my to-do list once I return home. I must admit that tapping the mechanical keys took me back to old-school keyboards before they eliminated key travel or felt "mushy". I look forward to playing with this.

I really like the "nub". This is the small rubber ball within the keyboard which can be used to control the cursor instead of the embedded trackpad. Both it and the pad feel smooth. No issues. It was a bit nostalgic to use the ball again. Everything feels premium.

I was invited to participate on a call with executives and developers from HP and System76. I jumped at the opportunity to learn more about this project. While they would not disclose too many behind-the-scenes details, I detected that more is to come. Typically, major manufacturers hide their Linux laptops within their store. HP created a separate domain just for this line. When

asked about future partnerships with HP, System76 CEO Carl Richell bit his lip for now. My impression, based on his mannerisms, was that we are going to see more HP machines which cater to the Linux crowd, specifically those who use Pop!_OS. That is something we should all be excited about. It may spur more competition and convince others to make the switch.

I asked if coreboot had been considered for the BIOS, and they said it had. However, there was not enough time to implement it before launch. I suspect we will see this in the future, and hopefully we will be allowed to flash our own BIOS if it surfaces.

My one petty complaint is the NVMe PCIe 3 drive. I would like to see an option for PCIe 4 drives which would double the speed. Don't get me wrong, the drive is blazing fast at 3000 MB/s, but I have been spoiled with my desktop's 7000 MB/s. To put

this in perspective, the traditional spinning disk drives might get you 150 MB/s, a standard SATA SSD drive might see 500 MB/s, and my 2018 MacBook Pro reached an amazing 2500 MB/s. This thing beats all of them. Most people would not notice the difference between 3000 and 7000, unless they were modifying and querying large data sets all day or editing huge 4K videos. However, I must remember this is a \$1000 machine. I can't have it all at that price.

The other things I would like to see are a microSD slot, Ethernet port, and second SSD slot, but I realize those types of features are being eliminated in favor of a smaller footprint. Again, I remind myself this is only \$1,000 and 14".

Bottom Line: This may be the best \$1,000 laptop you will find which allows a powerful transition to Linux Pop!_OS. I don't miss Apple or Windows. Well, I

miss the Apple Retina display, but not the telemetry. This thing is solid. I will report back once I have more real-world experience with it and the peripherals.

You can find more details at <https://hpdevone.com>. Currently, only US orders are allowed, but international options should appear soon. If you have a more specific hardware need or want to customize your own machine, I recommend visiting <https://s76.co/System76Unredacted>.

HP sent me a demo review unit as a loan. They are not a sponsor of my magazine, blog, or podcast and did not pay for this review. HP had no influence on the review and gave no direction for my testing ■



**MDR | XDR | INCIDENT RESPONSE | PEN TESTING
VCISO | WEB3 & BLOCKCHAIN | MANAGED SIEM
HELPDESK | IDENTITY MANAGEMENT
DISINFO MANAGEMENT**



REAL-TIME THREAT
DETECTION



REAL-TIME THREAT
RESPONSE



PROTECT YOUR ENTIRE
NETWORK



PEN TESTING AND
VULNERABILITY SCANNING



REDUCE YOUR IT/SECURITY
WORKLOAD



AFFORDABLE
PRICING

FORTIFY24X7.COM | (800) 989-2647 | INFO@FORTIFY24X7.COM

A FOSS SOLUTION FOR RECEIVING SHORT CODE SMS

By SWhopper

My big frustration with VoIP phone numbers is they don't receive every SMS message sent to me. For example, when opening a new social media account I'm asked for a phone number to help prove I'm a real person. My VoIP number is seemingly accepted but the SMS containing my one-time passcode (OTP) never actually arrives. Or I could be attempting to log into an existing account which requires 2FA via SMS but again, the OTP never arrives. There are no error messages, no explanations, just a silent failure somewhere in the system.

This issue is reported by many others online and is not limited to social media websites. I've seen it in relation to banks, utility companies, shopping websites, to name just a few, and with all of the main VoIP providers. Those same posts usually relate such failures to 'short codes', 5 or 6 digit phone numbers which websites may choose to send their automated SMS messages. Websites can block VoIP numbers from receiving their short code SMS presumably because they want the 'something you have' part of two-factor authentication to be a real SIM card in a physical phone. Less often it's the VoIP provider which has blocked the receiving of SMS from short codes instead, although all the popular VoIP

options state they support receiving SMS from short codes by default including MySudo, Google Voice, and JMP.chat (Twilio is the exception, but will enable it on request).

So what are the options for someone like me whom wants multiple reliable phone numbers which can receive all SMS including from short codes? I could carry multiple real SIM cards and constantly swap them in and out of my primary phone, or I could have multiple real SIM cards that live in multiple additional phones which I carry everywhere instead. But that is impractical and would likely lead me to giving up and handing over my primary phone number just to avoid the hassle. Not a great win for my privacy.

However I've found a more convenient solution which I'd like to share with this community. One which I've never seen published anywhere before. I hope that sharing it might assist anyone else in this same predicament. It only requires a primary phone (i.e. the one you likely already carry everyday), one or more additional phones running Android 4.4+, a FOSS application suite, and a widely-used encrypted communication protocol.

Project MAXS

Project MAXS (Modular Android XMPP Service) is a free and open-

source suite of Android apps which allow you to receive notifications and control Android devices remotely. As the name suggests it's modular meaning each functional element may be installed separately as required from the F-Droid app store. It leverages XMPP (Extensible Messaging and Presence Protocol) for communication which is a well-established, open, and optionally encrypted protocol with many great FOSS clients available on both mobile and desktop.

The MAXS suite includes optional modules like those for remotely reading GPS location, reading or controlling Bluetooth and WiFi, or running shell commands, as well as the mandatory XMPP 'transport' modules and a 'main' module for coordinating everything together. These optional modules might give some ideas on how you could repurpose an old Android device but my focus for this article is the 'SMSRead' module which enables reading and forwarding of incoming SMS messages.

I won't go into the full set of steps to get Project MAXS installed and configured. For that I recommend you follow the quick-start guide on their (unfortunately HTTP only) website: <http://projectmaxs.org/documentation/quickstart.html>. However in broad strokes the process is:

- Take a phone running Android 4.4 or higher,
- Add one or more activated SIM cards (dual-SIM phones work for this to double the numbers per device),
- Install and configure MAXS modules for main, transport, and SMSRead according to their documentation,
- Point MAXS transport at an XMPP address loaded on your primary phone,
- And you're ready to use real SIM phone numbers with any website including those using short codes.

Every SMS received on the additional phone will be forwarded to your specified XMPP account on your primary phone and the additional phone can be left anywhere with a stable power supply and internet connection.

Potential drawbacks and considerations

Verification SMS messages are obviously a target to attackers and this arrangement adds an additional transport step to the equation. Therefore this will increase the available attack surface. Furthermore while Project MAXS is open-source and its Github repo has been active for 7+ years it has not undergone any code audits and it's not popular enough to necessarily rely on a sheer number of eyeballs to ensure code quality. Finally Android 4.4 is also very out of date and has not had a security update in many years. So for all these reasons I wouldn't necessarily recommend using MAXS for any vulnerable accounts like banking, finance, healthcare, etc.

However, mitigations like strong authentication on your XMPP account and only signing up to a trusted XMPP provider help (I use <https://conversations.im> who also makes their own mobile app available on F-droid),

as will making sure to use the optional encryption features of XMPP from within MAXS transport. Furthermore, although MAXS is compatible with phones running Android 4.4, I would strongly recommend using a phone on a higher version to get more recent security updates. There are a surprising number of older phones selling second-hand for cheap online which also have recent patched builds of Lineage OS available. For these reasons I believe MAXS is suitable for less vulnerable accounts which require 2FA via SMS to a real SIM number.

I've yet to explore the other optional modules of MAXS, for example the SMSSend module opens up the possibility of both receiving and sending SMS from a real SIM number, which may be useful to those outside the US without access to affordable and simple VoIP options for additional numbers. Perhaps a topic for a follow up article. For now I hope this article will be of benefit to someone else out there. ■

Is privacy and security overwhelming? We can help.

Whether you are ready for a complete anonymous relocation with a full privacy reboot or simply need a one-hour call directly with Michael Bazzell, we can eliminate the frustrations encountered when trying to be invisible.



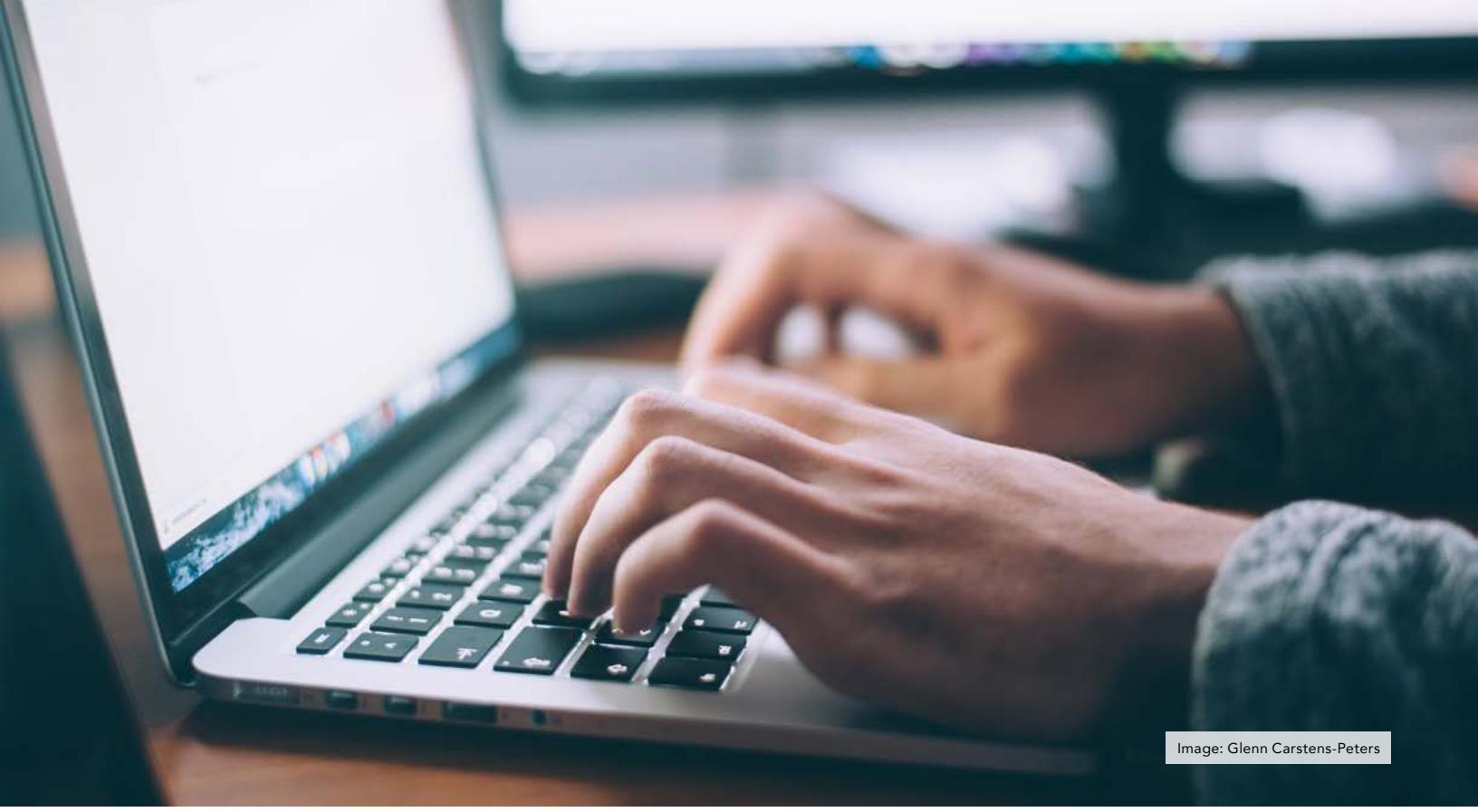


Image: Glenn Carstens-Peters

OPTIONS FOR YOUR DOMAIN

By C L

When I am looking online I see general confusion about how buying a domain works, the options available for registering, the rules around ownership and how you can take steps to protect your privacy. When you boil everything down you have three options available for registration that each come with their own pros and cons. This article talks about ICANN and for all those that don't know ICANN is an organization that sets the rules about domains.

1) Registering a domain with your real contact information and depending on WHOIS privacy protection. This method is likely best if your threat model does not involve trying to hide the domain registration from governments, but it will at least afford you privacy from entities with no legal reason to know who is the real registrant. Pros: This option is easiest, doesn't break the

rules, and affords you control of the domain. Cons: Less privacy than options 2 or 3. Options 1 and 2 are related as they don't rely on a proxy, but I want to keep them separate and present what I think is a better choice before option 3.

2) Registering a domain with an alias, incomplete, partially incorrect or completely incorrect contact information and depending on WHOIS privacy protection. I would argue this is the worst of the 3 options. Michael goes into some detail about this in his book Extreme Privacy and used a hotel address and partial name for the contact information of a domain. He is correct that he would not have violated any ICANN address rules for the length of the hotel stay, however, by not updating that contact address when it is no longer a contact address and by using a partial name it is a violation of ICANN's Registrar Accreditation Agreement subsection 3.7.7.1.

3.7.7.1 The Registered Name Holder shall provide to Registrar accurate and reliable contact details and promptly correct and update them during the term of the Registered Name registration, including: the full name, postal address, e-mail address, voice telephone number, and fax number if available of the Registered Name Holder; name of authorized person for contact purposes in the case of an Registered Name Holder that is an organization, association, or corporation; and the data elements listed in Subsections 3.3.1.2, 3.3.1.7 and 3.3.1.8.

3.7.7.2 A Registered Name Holder's willful provision of inaccurate or unreliable information, its willful failure promptly to update information provided to Registrar, or its failure to respond for over fifteen (15) calendar days to inquiries by Registrar concerning the accuracy of contact

details associated with the Registered Name Holder's registration shall constitute a material breach of the Registered Name Holder-registrar contract and be a basis for cancellation of the Registered Name registration.

RAA subsection 3.3.1 goes into more details about the personal information required here: ICANN RAA 3.3.1

- 3.3.1.1 The name of the Registered Name;
- 3.3.1.2 The names of the primary nameserver and secondary nameserver(s) for the Registered Name;
- 3.3.1.3 The identity of Registrar (which may be provided through Registrar's website);
- 3.3.1.4 The original creation date of the registration;
- 3.3.1.5 The expiration date of the registration;
- 3.3.1.6 The name and postal address of the Registered Name Holder;
- 3.3.1.7 The name, postal address, e-mail address, voice telephone number, and (where available) fax number of the technical contact for the Registered Name; and
- 3.3.1.8 The name, postal address, e-mail address, voice telephone number, and (where available) fax number of the administrative contact for the Registered Name.

They make no mention of demanding a real physical address only a postal address. I can't speak to if Namecheap words their demand for an address to be physical or not but it sure looks to me like ICANN isn't demanding a true physical address. Looking at the two registrars I have used they make no mention of needing a physical address. I don't see any reason this address couldn't be a PMB, UPS mailbox or PO Box. For option 2 this would likely be a better alternative to providing a temporary address or partial name as it is still technically correct contact

information. Is someone likely to cancel your domain registration for incorrect information? Probably not. Would they do it without asking you to update your information first? Maybe. Your registrar probably has bigger fish to fry but if you are going to give just enough information to argue you own the domain just go the small extra step to have correct contact information to mitigate the risk of domain loss. Using a VoIP number, email not used for other purchases, and a PMB should make you at least reasonably hard to locate but still does not break any rules.

3) Using a proxy domain provider such as Njalla. This isn't an endorsement for them specifically just using them as an example. This is the most extreme option if you are really trying to keep your information private. I want to specifically highlight this option will have varying degrees of privacy based on the history of the domain and payment information. If you have previously registered this domain with your true identity ever then the history will still be out there that can be found. If this is a new domain that has never been registered but you use a credit card, PayPal or other not privacy friendly cryptocurrency to register the domain there is potentially a digital trail back to you.

The major pro to using a proxy for your registration is that if this is a new domain, you use a reputable VPN or anonymous IP source, and you paid with a crypto that has some privacy protection built in or came from a source that doesn't have other transactions to tie back to you then you can be reasonably assured your identity can't be linked to the domain. Another pro is that the proxy may choose to ignore legal challenges from countries they do not reside in, and this could be a benefit you are looking for depending on where you live. The major con is that you really do not own this domain. If Njalla were to disappear tomorrow it would be difficult, expensive or maybe even be impossible to recover control of any domains you have registered with them. The other downside is you need to know how to use cryptocurrency and

how to obtain it without it being traced back to you. I wouldn't consider PayPal or credit cards a real payment choice if you are going for extreme privacy.

If you are planning on doing anything especially sensitive with this nuclear privacy option that can't be tied back to your identity, then you also need to ensure that everything you use the domain for also cannot be tied back to your identity. If you rent servers, host your own services or are just using it for email then you need to ensure that the provider you are renting services from also does not know your identity and if you are hosting your own stuff then you have to host at an IP address that also can't be traced back to you. Even if you are just using it for email then you have to take steps to not leak a true IP address on any account where an email address at this domain is registered or used. I do still think there is some privacy value to using a proxy for registering your domain even if you are not doing anything especially sensitive but it will vary based on your own history with the domain, the payment method used and how you use the domain.

In order to choose what option is best for your individual situation it may help to consider how much control you need over your domain and weigh it against how important it is that the domain not be tied back to your identity. You either need control of the domain more than privacy which would lead you to option 1 or you need privacy more than control which should push you to option 3. In my opinion option 2 really just isn't an option as it tries to straddle both of the other options and breaks the rules in the process, risking your domain registration being cancelled while affording little more privacy than option 1. If you really can't have just your real name with a VoIP phone number, email used only for the domain, and PMB associated with your domain at all then that should be enough to push you to use a proxy like Njalla.

If it is of dire importance that your name not be tied back to a domain it would be wise to consider if a domain is really needed for your purposes or if easier alternatives are available.

VARIANT DETECTED.

**Websites and graphics.
For businesses who
respect privacy.**

Be a variant.



Astropost

Astropost is the official design partner for this issue of UNREDACTED MAGAZINE. Need an ad designed for the magazine? We'll help you out!

APARTMENT LIVING

By Eugene Debs

When I sold my house in Wisconsin about five years ago I decided I would only be renting apartments in the future, so I needed to figure out which privacy techniques worked best for me as a renter. I first established a UPS Store address locally, providing a copy of my driver's license with the residential address of the house which I had just sold. I then changed the address with the Department of Motor Vehicles to be that of the UPS Store box address. Wisconsin allows UPS Store addresses on driver's licenses, though I'm told that not all states do. I then moved to another state with my girlfriend and we rented an apartment at a large corporate-owned complex. They wanted the usual information from me, including driver's license, which I provided as I could see no other options at that time. All utilities are in my girlfriend's name except for the cable/internet service, which is in my name (misspelled). My previous address they have on file is that of a Wisconsin UPS Store.

After living there for a while I became aware of license plate scanning, which is carried out routinely by police and other vehicles containing automatic license plate readers (ALPRs). Information on the vehicle, date, time and a photograph are uploaded to

databases, making it easy for vehicle tracking. Members of the public have difficulty obtaining this information, so I paid a private investigator to run a report for me on my vehicle. It showed that, on average, my vehicle was being scanned three times per month, mostly at this apartment complex. I'd like to know the legality of scanning vehicles on private property. When I decided to buy a new vehicle, I installed a retractable license plate shield on the front and I reverse my vehicle into every parking space, so my truck isn't tracked where I live or where I park. A better solution would be to keep it in a garage but that doesn't work for me right now. In the years I have lived in this other state I have kept a Wisconsin driver's license and plates. Despite states requiring you to change the address on your driver's license within ninety days of moving there, this seems to be something that is not enforced. I know several people who have never changed their address with the DMV, so I know I'm not alone.

After a while I decided to rent an additional apartment back in Wisconsin. The place I found was privately owned, and once again I found it necessary to rent in my real name, which I provided for background checking. They asked me for a copy of my ID, but I somehow kept forgetting to give it to them. When it came to setting up utilities, I decided

I wanted to be known as FirstName MiddleName. As Michael describes in his book, setting up internet service in a different name is surprisingly straightforward. Establishing power, water and gas services privately were much more challenging. Whomever I spoke to & however I approached the discussion, the utility companies needed my actual name to carry out background checks. I finally relented, gave them my real name, but insisted that the services be set up with my assumed name, arguing that I had been the victim of identity theft. They agreed. All bills arrive containing this new name and I have a credit card with my assumed name that I use to pay them. When I log into these accounts I only see my assumed name. I am working on the assumption that the utility companies have a record of my real name somewhere on file, but will only reveal it with a court order. This apartment complex doesn't require home owner's insurance, so I have chosen not to buy it, avoiding that privacy challenge.

Despite carrying out multiple people searches on myself, I have yet to find my name linked to either of these apartment addresses. I am very interested in learning how other privacy advocates get around the complexities of renting apartments without setting up LLCs. ■



Image: Ross Sneddon

MAINTAINING PRIVACY IN THE UNITED KINGDOM

By UK citizen

Michael Bazzell's books and podcasts have allowed everyone to understand how to lead a more private life. There has certainly been a United States slant to all of this, and rightly so as it is hard to speak with authority outside of one's own jurisdiction. However, I think it would be valuable for the community to share a perspective of privacy in some other countries – whether people live there, travel there or just want to compare approaches within the United States. I've worked in

various law enforcement departments and am a certified information security manager (CISM) with membership of the International Association of Privacy Professionals (IAPP). Here is a personal perspective of how to be more private in the United Kingdom, covering the context, differences to the U.S. and some of my top tips. I hope others can follow with other countries too.

Legislation and privacy context: The United Kingdom comprises four nations (England, Wales, Scotland and Northern Ireland). It had been part of the European Union (and the

predecessor of European Community) since 1973 until the Brexit Vote of 2016 and withdrawal from the EU in January 2020. The Data Protection Act 2018 is the UK's implementation of the General Data Protection Regulation (GDPR) and gives individuals rights over the use of their personally identifiable information and sets out rules that organizations in possession of personal data must abide by. The key principles, rights and obligations remain the same, but following withdrawal from the EU, the UK has the independence to keep the framework under review.

The UK has an extensive surveillance capability linked to its counterparts in the US and other five eyes partners, as highlighted by the Snowden revelations. Most recently, the Investigatory Powers Act 2016 (locally known as the Snooper's Charter) makes provision for both targeted and bulk retention of content and metadata, and enables the Government to require internet service providers and mobile phone companies to maintain records of (but not the content of) customers' Internet connections for up to 12 months. In terms of cameras, there is a national Automatic Number-Plate Recognition or ANPR capability (our equivalent of ALPR), London is third (behind two Chinese cities) in terms of CCTV cameras per person, and many police forces are trialing facial recognition to combat crime.

How is privacy perceived? People who tend towards extreme privacy are seen as a bit weird here in the UK, but perceptions are slowly changing. My favorite strategy when trying to minimize giving away personal data in person (apart from lying to government organizations) is to start off politely, then become firmer and finally state politely that I'm on a witness protection program and ask if there is any way that they can make an exception for me on the grounds of personal security.

What can individuals do to be more private in the UK? Of course, much of the guidance in Extreme Privacy translates easily for those in the UK and probably worldwide. I have a Linux machine with Firefox, behind a Protectl firewall running PfSense, Proton Mail, Proton VPN, KeePassXC, etc. However, there are a number of areas where things don't translate as easily, or at all, due to financial or legal reasons. While the UK probably has more legal protections in terms of our data not being shared (although we are subject to as many breaches), we don't seem to have quite the same ability as US counterparts to push the privacy limits in everyday life.

Identification card – The UK does not have national identity cards. The government introduced the CitizenCard primarily as a nationally recognized photo ID card and proof of age (for those of 16-18) to buy age-related products (e.g. alcohol). It only contains your full name, date of birth, photo and a unique reference number (that has no use elsewhere). It's not designed to replace a driver's license and full passport, so don't expect to hire rental cars, etc. However, it is open to all and I've used it to provide ID in hotels, and to provide official ID to some online sign ups, etc., (although you often have to insist that it is official ID).

Privacy credit cards – The UK's financial legislation means that sites such as privacy.com cannot issue cards in the UK. Additionally, alias secondary cards seem impossible to obtain.

VoIP numbers – MySudo used to be possible in the UK (albeit 5-10 times as expensive) but their site now indicates usage in U.S./Canada only. Abine Blur provides a free VoIP number forwarded to a mobile (cell) number (I'm on a \$55 for 3 year package).

Anonymous mobile service – You can buy basic burner Nokia 105 phones for £15 (~\$20) with 2G GSM capability, no camera, and no GPS. The cheapest anonymous service is Lebara (£5 per 30 days for 1000mins, 1000 texts, 1GB data), available for cash in convenience stores, with cash top-ups available too. Additionally, if you want to set up burner accounts for OSINT, you can buy Vodafone SIM cards in convenience stores for £1 without service (however you can receive SMS texts free for the period up to the expiry on card and/or six months whichever is sooner); I buy lots and rotate them through a burner phone. Lastly, if you want to keep a burner/emergency phone the mobile carrier 'Three' is cheapest way to keep active service (SIM cards cost £10 and you only need to make a transaction every 6 months for £0.10, so could last

for years), but this service only works on 3G phones.

Google Voice – I've not been able to set up Google Voice accounts (even using services such as TextVerified to create a phone number for registrations). My only success was commissioning someone to create two Google Voice numbers on Fiverr.

Bitcoin – In March 2022, the Financial Conduct Authority ordered all UK cryptocurrency cash machines (ATMs) to be shut down, or it will take action. I'm fortunate that I beat the deadline with enough to fund my Proton account, but the only remaining options are to use regulated means (online that requires Know Your Customer identification checks) or to find people willing to trade in person for cash.

Post office (mail) box – Royal Mail PO Box service allows people to have mail collected at their local delivery office or forwarded to their home address. Prices are around £300-£400 per year.

Travelcards – Transport for London operates buses and the metro across the capital. Since 2003, 'Oyster cards' were the predominant choice (prepayment cards that you touch/swipe on entry/exit). Clearly they capture/retain all journey details, but there has always been the option to buy them anonymously with cash. However, the predominant method is now becoming the mobile phone linked to Google/Apple Pay. There is speculation that Oyster cards may be phased out in future, although TfL (Transport for London) will still need to cater for those without bank accounts or do not wish to use contactless phone payments. ■

THE CASE FOR SWITCHING TO QUBES OS

By Wissam

I'm a jurist and work in the field of human rights. I started using Qubes OS in 2019. To make the switch, I watched several video tutorials online, read part of the documentation on www.qubes-os.org, and tested Qubes OS in a dual boot system, before getting rid of Windows once and for all. In one of his podcast episodes, Michael Bazzell's hinted that Qubes OS was not meant for the majority of people switching to Linux. I respectfully disagree. After 3 years of using Qubes OS, I believe that if privacy-minded people want to make the switch to Linux, it makes perfect sense for more than 50% of this particular population to switch to Qubes OS, using the system in a minimal and basic setup, rather than making the switch to a Linux operating system such as Ubuntu or Pop!_OS. The learning curve of Qubes OS is higher. Some tasks are more complicated but not insurmountable. But the return on investment of a minimal setup is exponentially higher. The documentation is detailed, and the community is responsive on the forum. And for day-to-day work, it's still Linux.

What is Qubes OS? In the simplest terms, it's the ability to run Linux, Windows 10 and other operating systems in virtual machines (VMs) simultaneously on top of an underlying, isolated and secure operating system called Xen hypervisor. The Xen hypervisor allows for strong isolation between virtual machines. The user never works in the Xen hypervisor. Running Qubes OS in a minimal and basic setup provides a less polished but equivalent work experience to running a Debian Linux distribution. By minimal setup, I mean using only one Qube or virtual machine for all tasks rather than splitting personal life, work life, cryptocurrency wallet, vault, etc. across several Qubes. In this minimal configuration, you would have 5 Qubes or virtual machines running: (1) sys-net Service Qube which controls the Wi-Fi and Ethernet and through which the Internet enters; (2) sys-firewall Service Qube through which the Internet passes; (3) the work Qube where you will do all the work, browse the web, and store your files; (4) sys-usb Service Qube to which all peripherals other than Wi-Fi and Ethernet are connected; and (5) the vault Qube where you store

and access the KeePass password manager.

These 5 Qubes install by default. You can ignore "personal," "untrusted" and other Qubes which also install by default. And you can choose not to install the Whonix machines during installation if you don't use Tor. Practically, you will browse the web in the work Qube, check email, edit and store documents, view and edit photos and videos, use Signal and Wire, make video calls on Zoom, Teams and Webex, etc.

This relatively simple setup holds benefits far beyond what any other Linux distribution or flavor can offer. Sys-net Service Qube, not the work Qube, is exposed to the Internet. I don't have to worry when I'm accessing public Wi-Fi in the United Nations facilities, in a hotel or at an airport. This prevents hacks using vulnerable Wi-Fi firmware from infecting other Qubes. USB devices connect to the isolated sys-usb Service Qube from which you can safely copy files to the work Qube. The work and vault Qubes are not exposed to the USB drive. You can right-click on a suspicious PDF or Word document

and choose to "View in Disposable VM" or to "Edit in Disposable VM." These commands will open the file in a virtual machine created for that sole purpose, preventing a potential virus from infecting the work Qube. If the latter does get hacked, your passwords stored in KeePass in the vault Qube are safe and inaccessible to the hacker, and your microphone and camera are disconnected by default from the work Qube and inaccessible to the hacker. Finally, the hack of work Qube will not survive a reboot because the system files are used afresh every time from the "Template".

In addition to virtualization, Qubes OS uses a "Template" system whereby every Qube will boot using a fresh copy of the system files, including the installed software. By default there are two Template Operating Systems to choose from, Debian and Fedora. During the Qubes OS installation process, I recommend setting Debian as the default template for all Qubes because the Fedora release cycle requires reinstalling a new Fedora Template every few months.

Templates are a very critical component in the system and are, therefore, disconnected from the Internet by default (About Templates: <https://www.qubes-os.org/doc/templates/>). Most software has to be installed in the template. To do so, you need to run Terminal in the Debian-11 Template, and run, for example, "sudo apt install libreoffice" to install LibreOffice in the work Qube. But wait! Aren't Templates disconnected from the Internet? Yes they are, but Terminal in a Template allows software to install via a proxy mechanism which in effect keeps the Template isolated from the Internet. Furthermore, when a Debian system update is available, the Template, not the Qube is what gets updated. The Qubes Update mechanisms runs a standalone update virtual machine which provides an additional layer of security for the Templates. This may sound complicated, but the experience is surprisingly streamlined.

Software available is that of the Debian repository. Snap and Flatpak

can be enabled. And for some software, I download the .deb package in the work Qube, copy the package to the Debian-11 Template, and then install manually using the terminal command – for example, "sudo apt install /home/user/zoom.deb". You can install software in the work Qube. But that piece of software will vanish upon reboot of the work Qube. There are exceptions, such as AppImage installers, which can install and run in the "Private storage" and will persist upon reboot. (How to install software <https://www.qubes-os.org/doc/how-to-install-software/>)

By default, USB drives and SD cards are connected to sys-usb, and the camera and the microphone are detached from all Qubes. Before every video call, I connect the camera and microphone to the work Qube via the Qubes devices menu. As for USB drives, they can either be attached to "work VM" via the Qubes devices menu. But if you want to prevent a potentially dangerous USB drive from connecting to a Qube, the drive's content can be copied from sys-usb across to the work or vault Qubes through the inter-VM copy mechanism.

Many users switching to Linux may decide, at a later date, to upgrade their privacy practices. With this in mind, a compelling reason to switch to Qubes OS is that it is incomparably flexible and privacy scalable. Your email can connect through one VPN, your web browsing through another, and Signal Messenger through third VPN – simultaneously. Or you could choose to route your work email via one VPN, and your personal email via another VPN route or provider. You can send a VPN connection from one provider through the VPN connection of another provider, thus building your own custom double-hop VPN. You can run multiple Signal Desktop accounts simultaneously, and even route Signal Desktop via Tor. You can install a sketchy software that you absolutely need, and run it in an isolated environment. You can safely open that suspicious website link via Firefox web browser running in a disposable, ephemeral, virtual

machine. And you can run Windows 10, Ubuntu, Fedora and Debian on the same machine – simultaneously.

Beyond this basic and minimal setup, I agree with Michael Bazzell that Qubes OS is challenging to use. Here are some of the challenges that I have encountered.

Finding the right hardware is challenging. I've wanted to use Qubes OS since 2018. However, it wouldn't install on my laptop until a newer Qubes OS version was released that included support for my hardware. Generally speaking, a two-year old laptop without a dedicated graphic card and with 16 GB of RAM should work fine with Qubes OS. (A non-exhaustive hardware compatibility list <https://www.qubes-os.org/hcl/>).

Some basic tasks and settings can be much more difficult to setup than on a normal Linux machine. Enabling a Blue Screen Filter is more complicated because it needs to be installed and configured in the underlying Operating System (Xen). Setting up a permanent Wi-Fi MAC address randomization in sys-net requires modification in the Template. The ideal VPN setup is to configure OpenVPN in an VPN Qube (A Proxy Virtual Machine) to which the work Qube connects (How to setup up a VPN in Qubes OS <https://github.com/Qubes-Community/Contents/blob/master/docs/configuration/vpn.md>). Modifying the screen DPI requires making changes in the underlying system and in the Template. Bluetooth is disabled so you can forget about wireless headset connection. The default menu system needs time to grow familiar with it. I personally access software using the Application Finder with the Alt+F3 shortcut. Software available in the Debian repositories is usually older releases, because that's Debian's approach to software, and is less polished than on Fedora, for example.

With all this in mind, I do believe that more than 50% of privacy-minded people who would want to switch to Linux are able to invest the time and effort to make Qubes OS work for

them, and will find my proposed basic and minimal Qubes OS setup much more rewarding.

A final advice for those making the plunge into Qubes OS – and some Terminal commands.

After a fresh install, and with Debian-11 being the default Template, I usually make the following modifications to get the system functioning smoothly. The system storage in the Debian-11 Template will need to be increased from 10GB to 20GB before installing software. The private storage in the work Qube (where your emails, documents, photos and videos are stored) will need to be increased from 10GB to 50GB or 100GB or to whichever size you need. This can be done in the work Qube's Qube Settings "Basic" tab. In the Work VM's Qube Settings "Advanced" tab, double initial memory to 800MB, max memory to 8000MB, and VCPUs to 4. This will allocate more RAM and virtual CPUs to the work Qube. In the sys-usb's Qube Setting "Advanced" tab,

set initial memory to 1024MB as this will make video conferencing smoother.

This is the command to install essential software in Debian-11 Terminal for "work VM" office (LibreOffice, PDF Arranger, Calibre) and multimedia (audio, photo and video editing, media import and management):

```
sudo apt install libreoffice vlc pdfarranger audacity kdenlive simplescreenrecorder calibre rapid-photo-downloader digikam gimp kodi
```

Copy/pasting text across Qubes requires an extra step. For example, to copy this Terminal command from the work Qube to Debian-11 Template terminal, select the text, then Ctrl+C followed Ctrl+Shift+C. This copies the work Qube clipboard to the Global Clipboard. Now to go the Debian-11 Terminal and Ctrl+Shift+V followed by Ctrl+V (or Shift+insert) to paste the text in the Terminal. And run the command.

Michael Bazzell recommends OnlyOffice. To install it, download

in the work Qube OnlyOffice .deb version (https://download.onlyoffice.com/install/desktop/editors/linux/onlyoffice-desktopeditors_amd64.deb), then move the installer to the Debian-11 Template (Right click then "Move to other AppVM"), and then run "sudo apt install /home/user/QubesIncoming/work/onlyoffice-desktopeditors_amd64.deb" in Debian-11 Template Terminal.

The Qubes OS project has made detailed instructions available on www.qubes-os.org/doc. YouTube videos in a dozen of languages include step-by-step installation information. The Qubes OS forum <https://forum.qubes-os.org> will have answers to most of your questions, and the community is very responsive.

Up until 2019, I used Ubuntu, Debian and Windows 10. But after 3 years of using Qubes OS, I'm not looking back. And I believe readers of this magazine and the broader privacy and OSINT community would feel the same about Qubes OS. ■

Are Trusts and LLCs overwhelming? We can help.

We believe all large assets should be titled to a Trust or LLC for privacy protection. Doing this correctly requires a lot of experience. We make sure your homes, vehicles, and any other assets which require titling stay out of your name. Contact us to reserve a consultation.





Image: Ludovic Migneault

READER Q&A

By UNREDACTED Staff

Do you have a question or need clarification about a privacy-related topic? Submit it to us for publication consideration at UNREDACTEDmagazine.com. If you have questions, other people are wondering the same thing! Please make sure your submissions are actual questions, and not vague statements with a "?" at the end. Here are questions from last month.

Q: What would be best practice/recommendations for Yubikeys? Should I buy 2? One to use and one for backup? I have several computers that I have, should I buy one for each computer? Which ones would you recommend?

A: I possess two. One is a backup. However, I do not believe you need one for each computer (unless you are worried about a physical forensic audit for evidence of usage). You should also

still create backup codes for any service which allows it. I prefer Yubikeys which are near flush to my machine, such as the 5 Nano and the 5C Nano. Once OnlyKey has small variants in stock, I will be testing those.

Q: We should try to use alias names for utilities whenever possible, but what if water is provided for residents by the city? Wouldn't it be illegal to lie to them?

A: I should first challenge your statement that we should always use alias names for utilities. I don't agree with that. Many utilities are government-owned operations and using an alias could be a crime. More important, there is really no need. The use of trusts and LLCs as the utility customer work great and do not involve deceit. If the utility requires a SSN or EIN, one can be assigned to the LLC used for billing. The only utility with which I consider an

alias name is internet access.

Q: You use Authy. I have used Authy for quite a while and cannot deny its utility but I don't really like its "lock in" so have started migrating over to KeePassXC's TOTP. As you have stated a preference for KeePassXC and that you manually move your kdb files around I was wondering if there was a reason you did not use KeePassXC's TOTP option?

A: I don't like all of my eggs in one basket. If my password manager possesses my passwords and the 2FA tokens, then it is only one-factor authentication. That is a bad idea to me. Also, I don't agree with the internet mobs who insist Authy is locking us into their free service by not showing us the seed code. I do not believe they store the seed code in a way in which THEY can see it at all, and therefore cannot easily export it to you. This could be a

good thing, but that does not fit the anti-corporate narrative people desire. The article in this issue about Authy proves you can absolutely export your seed codes while logged into your own Authy account, so you have the option to leave. If someone is mad that Authy cannot see or easily share your seed codes, then they must also be mad that Signal won't allow you to see your secure messages within Google Voice and Proton Mail isn't accessible from Yahoo. Apples and oranges, but I think my point is clear. Know your options and make the best decision for yourself.

Q: Is there any value to changing house to LLC/trust and staying in it? Your name would show up as the previous owner but?

A: Yes, but it is limited. It might prevent future privacy invasions, but it will never completely erase the history of the home. It would stop new data collection from county records.

Q: After enabling forwarding from Gmail to Proton Mail, as for the outgoing messages from the Proton Mail account, are there any workarounds to show the sender as the Gmail account, without logging in to the Gmail account again?

A: No, and this is by design. I use this as an opportunity to convince others to send email to my secure account and not a Gmail address.

Q: How to securely make offline backup USB storage for Android phones, with no availability of VeraCrypt?

A: If you are a macOS user, you can encrypt external drives with FileVault. Linux users can use LUKS. Any drives which I only use with Linux are encrypted with LUKS from within the Disks application.

Q: Can I stop my company from sharing my salary/payment data with Equifax? I work for a very large multinational employer, and recently found they are one of the many who share extensive paycheck/salary data with Equifax's 'The Work Number'. I am working on limiting The Work

Number from being able to share my data, but is there any way I can request my pay data be excluded from being sucked up into this system in the first place?

A: You can absolutely make the request, but I wouldn't expect them to help you. I do not believe we will ever see employment privacy again. Self-employment under an invisible umbrella LLC has its advantages.

Q: Extreme Privacy and this magazine have stressed the importance of a trust for anonymous home purchases. The book mentions that one strategy is to find an attorney willing to sign on as a trustee, which seems like the most secure option. Do you have advice on finding the right attorney? Setting up consultations and advertising my intention to buy a home anonymously seems to defeat the purpose.

A: I guess I would first ask you why you feel that speaking to an attorney defeats the purpose of your intention to buy an anonymous home. People do this all the time. If you don't want your attorney to know you are buying a home, even with attorney/client privilege, I am not sure how to advise you. Initial meetings do not require the address of your purchase. They might not even require your name. How do you plan on working with a real estate professional without letting them know you are interested in purchasing a home? I think you need to focus on your scope and threat model more. As for finding the right attorney, I focus on estate planning offices.

Q: I live in the US and am interested in purchasing bulletproof clothing, or at least a bulletproof backpack. Do you have recommendations? Or perhaps my money is better spent elsewhere, and I'm just reacting to all the news in regards to mass shootings and increasing crime? I'm a smallish female, travel to "big cities" and have little to protect myself save a tactical pen. I have had recent run-ins with the homeless in downtown areas and feel increasingly vulnerable. Thank you.

A: I am no expert in this area, but I think that most bulletproof clothing and apparel is a scam, especially anything affordable. Some may stop less powerful rounds, but only if they are properly positioned during an incident (which would be unlikely). I have not seen any bulletproof backpacks which can stop rifle ammunition, which is your largest threat. Maybe a reader has better news to offer.

Nick's unsolicited interjection:

Don't cheap out on body armor. There are some restrictions on sales based on your location, criminal history, and other factors, but here are a few places willing to sell body armor to unrestricted civilians: [T.Rex Arms](#), [SKD Tactical](#), [Safelife Defense](#).

Q: As a student, privacy is often an extremely difficult goal to strive for. Keeping this in mind, how can I maximize my privacy when faced by inflexible digital policies from my educational institution, e.g., mandatory provision of personal details to be stored on their (most likely insecure) web server, online applications, downloading apps like Teams/a VPN?

A: I feel we could dedicate an entire issue to this topic, and I already presented many ideas within my book. The short answer is virtual machines, a good VPN, a masked mailing address, browser-based tools instead of app-based installations, and an overall resistance to complying with their invasive demands. Maybe someone else will expand a full article on this.

Q: Will you be using your Linux machine and OpenOffice to self-publish your books in the future?

A: I can say two things. The document used to generate the content within this issue was created with OnlyOffice on a Linux machine. However, the final layout was conducted by Nick within Adobe InDesign. If I publish another book, it will absolutely be created on a Linux machine. I look forward to the hurdles. ■

BOOK UPDATES

By Michael Bazzell

Last month, I discussed my transition to Pop!_OS on my personal Linux machines, which has prompted many readers to offer their own feedback. Most has been positive and many people have also made the switch. One issue which keeps surfacing is virtual machines.

In my books, I explain the creation of Ubuntu VMs for both OSINT and privacy. Many readers of the OSINT book are wondering if they should make the switch to Pop!_OS for their custom OSINT VMs while Extreme Privacy readers wonder if they should abandon Ubuntu within their personal VMs. My short answer to both is "No". The longer version follows.

I still prefer Ubuntu for my OSINT and privacy VM's for two reasons. First, stock Ubuntu is much more popular than Pop!_OS and I like to fit in with the masses when I am conducting an investigation. Submitting queries from a Pop!_OS VM may look too unique for my threat model, but this is probably unjustified caution. Overall, my VMs are all Ubuntu running on a Pop!_OS host.

However, the main reason I keep Ubuntu for VMs is simply function. All of the tutorials and scripts from the 9th edition of my book Open Source Intelligence Techniques were written specifically for Ubuntu, and updated online specifically for Ubuntu 22.04. Would they work for other Debian-based operating systems? Mostly, yes. However, you will run into path issues with the Firefox template and installation issues with some applications, such as EyeWitness.

I don't want to maintain two sets of instructions, and the book is published using Ubuntu. Therefore, I will stick with that. If you like to tinker, I suspect you could adapt 95% of the applications, scripts, shortcuts, and configurations included in the book to Pop!_OS or other flavors. Please let us know if you do, I would like to see your process. Better yet, submit it as an article to the magazine. I am sure others would be interested. ■



This magazine serves as a compliment to the podcast, which can be found at IntelTechniques.com. Below are summaries of the episodes from last month.

264-Back to Basics: Linux 1

We return to the basics and start with Linux. Whether you are ready to make the switch or have been using Linux for years, there is something here for everyone. We summarize and expand on many of the Linux tutorials presented in issue 002 of UNREDACTED Magazine.

265-HP Dev One with Pop!_OS

I get my hands on the new HP Dev One with Pop!_OS pre-installed and offer a full review, which is very similar to the article presented in this issue.

266-The Power of the Sole Proprietorship

I explain the privacy features of a sole proprietorship with corresponding EIN.

267-macOS Privacy & Security Revisited

After a lot of talk about Linux, I revisit privacy and security considerations for macOS machines.

LETTERS

By UNREDACTED Staff

"Page Numbers" by https443

Thank you for putting together a great publication! My one small critique: if I'm telling a friend to "Check out the article on page 20," it's a bit confusing. To resolve this issue, simply make the cover "page 1".

Editor's Note: We are definitely learning as we go. I had never considered the cover to possess a page number, but I see your point so that the visible page number in the magazine matches the page number presented in the PDF reader. Fortunately, our new layout since issue 002 corrects this.

"Cat Stickers" by Anonymous

I'm a relatively new member of the privacy community, but I just wanted to share a small success story and a potential helpful tip to others who are trying to convince their friends and family to switch to secure communications. I've been able to convince about 80% of the people I personally communicate with on a regular basis to switch to Signal by promising to create custom stickers for our group chats. All of my friends and I now have custom stickers of our cats, faces, etc. to use in our chats which has sold most of my group on switching to Signal. It only took a few minutes to create each sticker, and it was fun to do as well! I don't know 100% if the stickers are publicly shared, but I think each sticker pack has to be manually shared in order to be used. The following link has a tutorial on how to make the stickers: <https://www.androidcentral.com/how-create-custom-stickers-signal-messaging-app>.

"Privacy vs. Security" by Sam Howell

Privacy: The state of being secluded from the presence, sight, or knowledge of others.

Security: The condition of not being threatened, especially physically, psychologically, emotionally, or financially.

These are broad terms; and the above definitions don't adequately cover all of the associations and interpretations we've come to understand of both subjects. Beyond our definitions of the two subjects, the difference between them is also indicated by the fact that many people, myself included, tend to write about one more than the other. Privacy is a basic human right and should be recognized as such the world over, but this isn't the case. Security, on the other hand, is too often the irresistible end governments and companies promise us to justify their questionable means – too often violations of our privacy.

I can't help but harbor an enduring distrust of a certain four-letter agency since it came to light they systematically spied on (and no doubt continue to spy on) countless innocent citizens at home and abroad, thereby violating their right to privacy. However, I have no doubt that same four-letter agency keeps us all safer in ways most of us are oblivious to. Crucially though, at least in my mind, this will never justify the above-mentioned violation of our right to privacy.

Which is more important? Neither, objectively. Is it more important that you lock your front door at night, or that you draw your curtains when you're getting dressed? That depends on all sorts of subjective factors related to your threat model, such as what the crime rate is where you live, who you are, what you do, what person, corporation or government you've pissed off and, more generally, how you feel about someone potentially looking into your private life without your consent or knowledge.

If one thing is clear, it's that to get by as the 'average' netizen in today's connected world without being screwed by some person, corporation or government, you need to know at least the basics of good privacy and security practices. That means worrying less about how to pull the wool over the 14 Eyes (which you'll probably never be able to do), and more about how you can avoid becoming low-hanging fruit for phishers and other malicious actors. After all, with receipt of one poorly-written email and with one careless click, your whole life could be upturned.

The value and importance of having instant access to so many privacy and security tools and other resources nowadays can't be understated. In just a few minutes, you can significantly improve your level of privacy and security online. This is something to cherish and fiercely protect.

"Strategic Privacy Goals" by Tom Cody

I began an interest in the privacy field after leaving my post-military law enforcement career for the private sector. Soon after being a civilian again, I began using a tailored Red Team approach in consulting with current government employees in finding their personal information online that could be used against them and with cleanup. After realizing the difficulty in relaying information security practices through a client's mental lens, I started to over-simplify their needs to them in three categories of what needs to be protected; this being in lessening the ability for an adversary to locate, harass, or defame them.

Locate: Having had several higher-risk positions in my previous profession, I learned the importance on keeping a close watch on how easy it is for someone to find where one lives. Not contaminating new residences would be among the best of practices. In many cases, home ownership records, previous utility bills, and just plain ol' years of mail have already done a great disservice. While there are still ways to minimize the ease of access to records online, good hygiene with a new address is always a great opportunity to start fresh.

Harass: Next up would be current contact information that would allow one to directly reach you or loved ones. From people search sites to social media, there are a million ways someone can find direct access to you. If you cannot opt-out, delete, bury, or otherwise remove the data, at least be aware of what is out there and where it is. A good habit is to do periodic self-audits to locate which contact information is discoverable and where.

Defame: Defamation is another focus area in my work. This includes potentially damaging things such as, controversial social media posts or comments, inappropriate pictures, or just current employment information that allows someone to channel their hostility against you to straight to your highest levels of management.

Remember, the nexus to family and friends, or oneself to an organization is often benignly published!

Lastly, one thing to address is the level of criminal sophistication of threat actors. Many of the techniques I have used over the years are more intended for lower-level and less tech-savvy threat actors. As the threats begin ascending into the tech-savvy or higher-level actor, it requires much more maintenance and care to keep information safe. For those who continue to protect themselves, maintaining online self-awareness and a healthy practice of monitoring, opting out, deleting, burying data, and using disinformation, will help make them a harder target. Being a harder target means that it takes more time and effort to be focused on, ideally, resulting in the threat moving on to someone else.

"Finding the fastest Proton VPN IPs for a pfSense router" by Flaming Gorge Utah

My home is hundreds of miles from the closest Proton VPN server, so I've always wondered if I'm connecting to their fastest VPN servers. Previously, I'd go to Proton's website and find the closest city's VPN servers, choose a few servers with little load, and then do a DNS lookup to find each server's IP. Finally, I'd input a few VPN IPs into pfSense using the steps in Extreme Privacy. Recently, I discovered a simpler way! I download the Proton VPN client to my computer and connect to my home network that isn't behind a VPN. Within the Proton VPN client I select "Profiles" and then "fastest". I then write down the VPN IP address that I connected to. I disconnect and reconnect several times to see if there are different servers that are fastest at any point in time. Finally, I log into pfSense and input these IP addresses (again, following the steps in Extreme Privacy.) With the Proton Unlimited plan and this VPN server-finding approach, I'm consistently getting <30 ms ping, 1 ms jitter, and over 100 Mbps upload/download (using the 4 port Protectl Vault). This speed and latency is good enough for multiple video calls without lag.

"Comments on: The Linux Lifestyle" by Reginald

Six months ago I purchased my first System76 laptop, running Pop!_OS, with every intention to immediately unbox and plug in, waylaying my Windows laptop (a decent Sager build) and making the intrepid and oft misunderstood leap to the Linux life. That taught me that old and reliable things are very hard to give up. In the following three or four months, I would devote an hour or two every three or four nights to playing on my newest toy, slowly learning some bash, figuring out the file structure, and frankly spending more time on Firefox looking up how to get things done on this or any other Ubuntu fork. The process had been slow, and I was still spending 95% of my computing time on the trusty Sager. I resigned myself to the dishonest hope that I would pick a day to make that inevitable switch. I arbitrarily picked the day that Windows 10 would stop being supported by Microsoft. I chose this date because I needed a tidal force to pry me out of the Windows bear trap. No other motivation worked, really; I am well-versed in Windows, and I know how to mess with the built-in firewall, registry, and various open source apps to make it do what I needed. It all just... worked.

I could double click a file and it did what I wanted. Every executable just ran its course when I needed it. Every app was made for me and my operating system. All the gadgets and peripheral extras were plug and never-fails play. The world turned, and I wrote whatever PowerShell function I needed. Friends who forgot their passwords or needed something fixed would come to me, and I would open that command shell and do some magic to give them their own world back. Family would break down in tears over not having their Wi-Fi password handy, and I would "give them their internet back." Time and again I heard that voice - "Why switch? You're fine with what you have. Why force it? Stay comfortable and in charge..."

Three days prior to me typing this, I read the first installment of "The Linux

Lifestyle: Switching to Pop!_OS". As is the usual routine with a Bazzell book, podcast, or article, I stopped everything and started comparing settings, tools, and digital postures. This time, it wasn't to make sure I was at the cutting edge of some privacy worry or counter-intelligence threat. I instead was busy testing NextDNS, scanning the man file for Eddy, loading my music onto Kodi, and disabling hardware acceleration in my newly hardened Firefox instance (having that setting active will mess with the way Firefox renders on my particular build). I spent more time on the System76 box than ever before, and now I am drafting this email on what is today my daily driver for computing.

This article series kicked-started my motivation to ditch Windows10 and make good on my recent investment. I am quite excited that this will be a regular feature in UNREDACTED, and I expect I'll learn heaps, as I have no one else here who can tutor me in Linux. With the same dedication to learning and conquering Windows applied here; I will no doubt get back into that comfort zone. Many thanks for reading the room, as it were, since I believe many of us have made this switch and might be struggling as I was. Alright, let's learn Bash.

"Regarding Wi-Fi Geolocation Concerns" by Kevin

Regarding Privacy Mike's Wi-Fi Geolocation Concerns article in issue 002. Perhaps changing the SSID of the Wi-Fi network to include the text "_optout" and "_nomap" would prevent Microsoft and Google from collecting information about his network. That is what I do based on several articles on the topic.

"Reader Comment" by Mike

Regarding the "Eulogy for the iPod Touch" article in issue 002, I wanted to point out that even if you can't buy a new iPod Touch, an old iPhone still works great for the same purpose. I have an old original iPhone SE which I use for accessing MySudo at my house. I have the Proton VPN always on, and it's connected via Ethernet through two

Apple adapters. The first adapter is a lightning to USB adapter, then I have a USB to Ethernet adapter after that. This setup is fantastic, and I can leave it at home all the time. When I leave home, I simply unplug the lightning cable from the phone and voila. With no SIM card in the phone and Wi-Fi and Bluetooth turned off; I do not believe the device is traceable. Am I wrong in this assumption?

Editor's Note: The point of the iPod Touch was that it possessed no cellular capability. It could not connect to any cell tower. A traditional iPhone can (and does) connect to cell towers even without a SIM card, sending data about the hardware identifiers. If your device ever enters non-airplane mode, which is common after an OS update, you could absolutely be tracked.

"Feedback" by Anonymous (two letters from two readers)

First Reader Submission: I'm a regular listener to the podcast and reader/user of Extreme Privacy third edition. When you announced UNREDACTED Magazine and your intention to crowd source the content, I was pretty excited about some ideas for articles to contribute. But I haven't. Why? Well, I just didn't have time before the first edition. And then when you announced its release, I downloaded it and... yeah, no. I am just not into the large-format, columnar PDF layout. I heard your explanation of why you wanted to do it that way, but it just doesn't work for me. At my desk, I can't fit a whole page at readable size, so I have to scroll down to the bottom of the column and then scroll back to the top. It interrupts the flow of reading and adds to my carpal tunnel irritation. Away from my desk, I'm on my phone. I have a large-screen phone precisely because I do a lot of reading on it. But again, the PDF format is a real pain. I guess I could print it, but toner is expensive... and printing is just so last-millennium. Lame. So it's hard for me to feel motivated to write articles for a publication I'm not going to read. I'm not mad ... I really won't be reading or contributing to the magazine in this format.

Second Reader Submission: So I downloaded your new "magazine". I wanted to let you know it is not 2002 anymore. People don't buy or read magazines. Please stop with the time making things pretty and focus on giving away content in a better format immediately online. If the next issue is in PDF then I will not waste my time downloading and I will encourage others on Reddit to avoid it. You have been warned.

Editor's Note: I have learned many lessons in the past 20 years creating content, and I want to share three of them here in case it may help others who are considering the same. **First**, you can't please everyone. When I publish print books, people are mad they are not available in PDF. When I give away a free PDF magazine, they are mad it is not in print. I have learned to accept that some people will never be happy, and that is OK. **Second**, I have learned that when most people say they do not like your product and will not be reading it any longer, that is not true. They will keep reading and may stay mad, but they never go away. I have a listener of my podcast who tells me every week he is never listening again. Guess what? He still listens every week. I suspect these unhappy readers will see every word in this issue and all of the words in the next issue. **Finally**, the biggest lesson I have learned is to stick with your gut and make whatever you like. I write the types books which I wish I could purchase, create podcasts I would want to listen to, and publish a magazine which I find beneficial. I don't change the recipe because of a small number of complaints. If you create things you like, nothing else matters. People will either join you or they won't. Either is fine. Don't sacrifice your goals for a handful of upset people which are incapable of being satisfied. If you are reading this and are upset at my response, I offer full refunds of the retail magazine price. ■



Image: Lucas Davies

AN ANONYMOUS CROSS-COUNTRY ROAD TRIP

By LibreWriter

It was autumn and I found myself yearning for another adventure. Flying and the recent global health crisis had made air travel and international destinations extremely unappealing to me. So if I wasn't going to fly somewhere or leave the country then it seemed a good old-fashioned American road trip was in order. Coast to coast! Yes, that would suffice.

At the time I was just getting started with the deeper, more nuanced implementations of privacy. I still had an iPhone and a vehicle with onboard telematics. The kind of telematics that provided an in-vehicle Wi-Fi hotspot and a smart phone app to view basic car details, locate the car, and also lock,

unlock the doors, and remote start all from the press of a button on a phone. Those things would certainly need to be addressed.

I was sold on GrapheneOS and the Google Pixel 4a as my ghost phone setup. Sadly, the Pixel 4a had been discontinued on the Google website and none of the local stores carried the device. I waited for months and the inventory was never replaced. So I begrudgingly surrendered to the fact that my spyPhone would have to come along with me on this trip.

The phone was using a SIM card registered in my legal name and I was not yet in a position to make changes to that. The phone number needed to be ported out, a more anonymous SIM

card would need to be acquired, and it seemed pointless to put that effort into an existing iPhone so tightly tied to my real persona. I did lock the phone down as tightly as possible. Deleted all non-essential apps, disabled all location sharing, and kept the phone in airplane mode nearly the entire drive.

It wasn't enough, in my mind, to just use airplane mode so I would remove the SIM card while the phone was connected to the cellular network thinking the SIM ICCID would no longer be associated with the phone IMEI. When I needed to use the phone, I made sure to never make a call or use the cellular data unless I was a minimum of 10 to 20 minutes (at highway driving speeds) away from where I would sleep

at night. And with no SIM card installed, I felt more comfortable using the phone on Wi-Fi when necessary.

With my spyPhone offline so much, how did I navigate? Before departing on this trip I wrote down the highway information and exit numbers on a small piece of paper and just read the road signs as I journeyed along. It's a pretty novel idea, I know! And you'd be surprised how fast a cross-country drive can go by while looking out the window without a GPS "time-remaining" painfully counting down to zero.

The vehicle telematics, though, is a tricky issue. Some vehicles have the TCU (Telematics Control Unit) hardwired into other car systems like the airbags or seatbelt alerts making it impossible to disable telemetry. And it seems most new vehicles sold today do not have the option to exclude telematics equipment. Fortunately, in my car, I found the TCU on a dedicated fuse and removed it from the fusebox. Much to my surprise (and enjoyment), there were no adverse effects with any vehicle systems.

I confirmed the Wi-Fi hotspot no longer worked; the car couldn't even see any cellular networks. Next I tried the smartphone app – errors, perfect. Even the vehicle screen confirmed it was not able to connect to any networks. It's possible this next part was overkill, but I then located the TCU hardware box inside the car and detached all the cables from the it. Finally, an invisible car! (Ghost car in a trust or LLC is the final step).

Next, using one of my alias VoIP numbers as explained in Extreme Privacy, I started calling around to hotels and motels to discover just how anonymous I could really be. Good behavior, polite manners, and friendly attitude went a long way here and also during check-in. My experience was that motels were almost 100% possible to be anonymous when planned ahead, and sometimes you have to make a few calls to different establishments in order to find one whose policies resonate with your needs. Hotels were more problematic overall so I simply

avoided them. Each motel I stayed at was locally owned instead of run by a big corporation. This is an important distinction because small, local owners were able to do whatever they wanted when working with me, without the requirements from corporate. And a nice incentive to motels is that you end up supporting a local economy instead of another big brand chain.

Motel reservations were made using an alias name and billing info with a VoIP phone number; a different alias and billing for each reservation (thank you Privacy.com). During the reservation calls I said, "I'm traveling with a friend who is driving separately. If they arrive before me, what do you need to let them check-in without me being there?" "Just call us the afternoon of your arrival and give us the name of your friend." That became the single determining factor in who I chose to book a night with.

Why? Because I wasn't traveling with a friend, I was traveling alone. So each afternoon I called the motel and told them my "friend" would arrive before me. The friends "name" was my legal First name and Middle name as it appears in my passport. Not the most anonymous method, but it was acceptable to me. And I would've acted aloof and said the missing last name was an accident had I ever been asked about that, but I wasn't.

Arriving at the first motel, I gave the alias name to check-in, not my passport listed First and Middle name. No ID was asked for, I paid cash, went to my room, and left the next day. The second motel I gave my alias name, was asked for my ID but acted clumsy and said "I must have left it in the car", they continued checking me in with no ID, I paid cash, and went to my room without incident. Had ID been enforced I would have found my passport and used the "friend" name as my pre-arranged backup option. The third motel was like the first. No ID was asked for, cash was paid, and I was once again a ghost.

Anybody that asked for a license plate number as part of the check-in, I simply made a "mistake" and mixed up

a couple of the numbers. For example, if the license plate number was 1234567, I would accidentally switch two numbers and use 1243567. An honest mistake for any weary traveler.

Using the internet at night was straight forward and simple. Power up the GL.iNet travel router, connect it to the motel Wi-Fi, enable VPN, and fire up the Linux laptop. There was never a time while traveling coast to coast, in either direction, where the travel router did not work. I used it at motels, hostels, campgrounds, libraries and each time it worked flawlessly.

Traveling only by use of cash was equally no issue at all. The habit was to take as much or as little as needed from an ATM in a location which I was immediately leaving. Never once did I make a withdrawal from a town or city where I was currently sleeping or had previously slept in. Motels, gas, food, tolls, all paid for anonymously with cash.

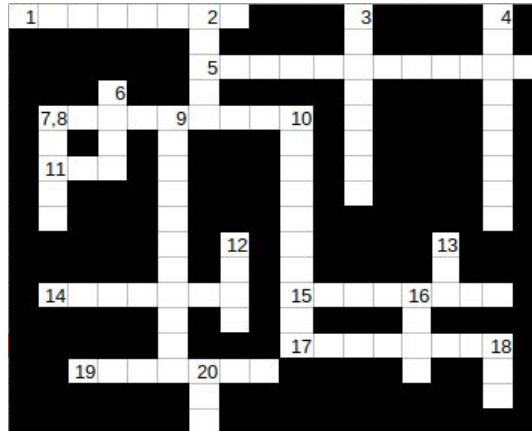
Arriving at my final destination I was able to keep my name and location hidden with the help of some friends and accommodations they could provide. All without needing to disclose to them my wishes for privacy and anonymity. That's what friends are for.

As you can see, it wasn't perfect but it was a darn good first try if I do say so. I never used my real name, never used a credit card or debit card, the car wasn't transmitting my every move, and my phone was disconnected from the mobile network nearly the entire time. You would've known me only by my occasional mobile phone tower pings, sporadic ATM withdrawals, and being seen by the security cameras that seem to occupy nearly every inch of public spaces.

Stay tuned next month for Part 2, the return portion of this cross country drive, as I refined my techniques, further reduced my digital footprint, navigated anonymous car repairs and unplanned motel stays, including everything I learned about camping like a ghost. ■

PRIVACY-THEMED PUZZLES

By Anonymous



Across:

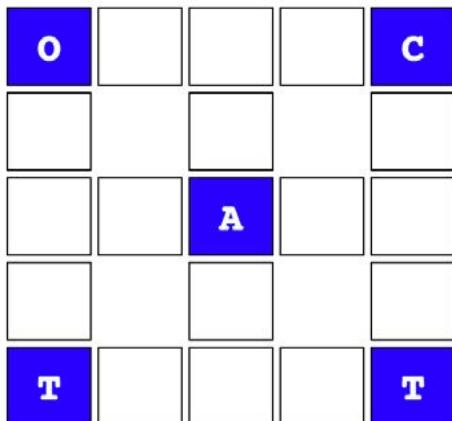
1. Previously saved state of a virtual machine
5. Strategy for isolating networks
7. 1 upper, 1 lower, 1 special ____
11. Malware enabling access from afar
14. Download to patch vulnerabilities
15. ____ mode
17. Enhanced privileges
19. Computational threat to encryption

Down:

2. Operational security
3. A lure
4. A form of authentication you are
6. ____ your hash
8. KeePass shortcut to copy username
9. Who done it
10. Common form of malware
12. Probability a threat will exploit a vulnerability
13. ____ 256
16. Data loss – no exploits required
18. Phone book for web sites
20. Successor protocol to SSL

Security Word Puzzle

Michael J. Ross



The objective of this puzzle is to discover the six words — all related to computer and network security — that fit in the above puzzle. Three of the words are horizontal and the other three are vertical, with overlap of some shared letters. Several of those letters have already been added to the puzzle to get you started. Here are the remaining letters to complete the puzzle:



CHUCKLES

By Annyong



FINAL THOUGHTS

By Michael Bazzell

Thanks for making it to the end of another issue. I can't convey my level of satisfaction with this new project in words. I still get excited each time a new submission reaches our group inbox. I finally feel like this privacy magazine concept will continue and grow, thanks to you. Please consider spreading the word or sending someone a free copy. I can't wait to see what this looks like in another year.

AFFILIATE LINKS

If you would like to support this free publication, please consider using the following affiliate links. If you plan to purchase any of the items below, or other items from the vendor (such as Amazon), the following links provide a small financial contribution to us without costing you anything extra. We see nothing about you or your order.

Extreme Privacy Book (Amazon): <https://amzn.to/3D6aiXp>

OSINT Book (Amazon): <https://amzn.to/3zoMZpZ>

ProtonVPN VPN Service: https://go.getproton.me/aff_c?offer_id=26&aff_id=1519

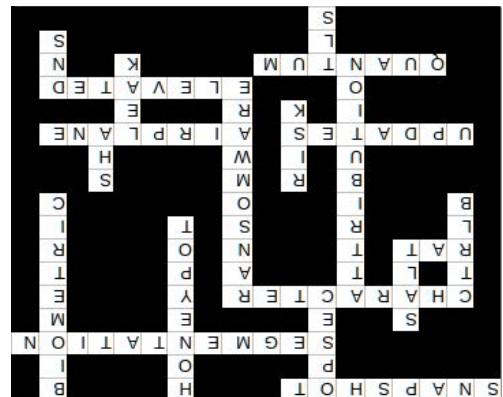
PIA Dedicated IP VPN Service: <https://www.privateinternetaccess.com/ThePSOSHOW>

ProtonMail Encrypted Email: https://go.getproton.me/aff_c?offer_id=7&aff_id=1519

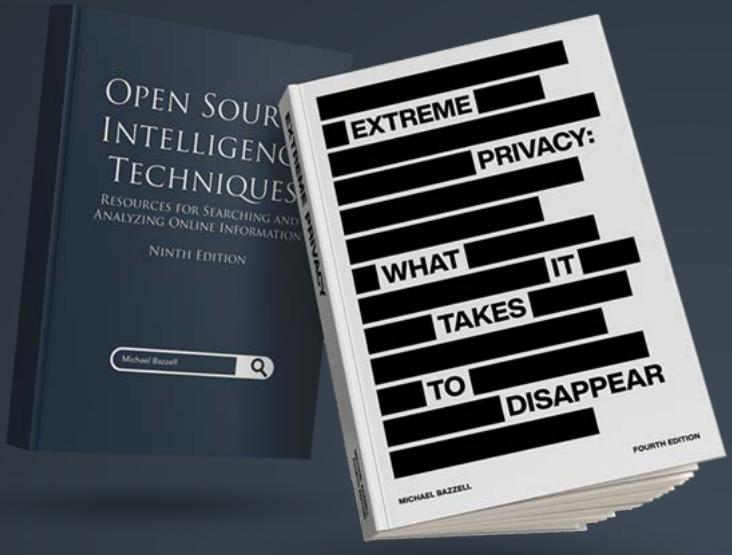
Fastmail Business Email: <https://ref.fm/u14547153>

SimpleLogin Masked Email: <https://simplelogin.io/?slref=osint>

Silent Pocket Faraday Bags: <https://slnt.com/discount/IntelTechniques>



New 2022 Privacy & OSINT Books



- ✓ Hardcover & Paperback
- ✓ New & Updated Content
- ✓ 500+ Pages Each @ 8.5 x 11
- ✓ Our Full Playbooks
- ✓ Supports This Free Magazine

Order at IntelTechniques.com

OSINT & Privacy Video Training

90+ Hours of Video Training | Optional OSIP Certification

Register at IntelTechniques.net

Open Source Intelligence
Video Training + Certification

\$899.00

