

[REDACTED]

[REDACTED] **EXTREME** [REDACTED]

[REDACTED]

[REDACTED] **PRIVACY:**

[REDACTED]

[REDACTED]

[REDACTED] **MOBILE** [REDACTED]

[REDACTED] [REDACTED]

[REDACTED] **DEVICES** [REDACTED]

[REDACTED]

[REDACTED] [REDACTED] [REDACTED]

[REDACTED] [REDACTED]

[REDACTED]

# EXTREME PRIVACY:

## MOBILE DEVICES

MICHAEL BAZZELL

EXTREME PRIVACY:  
MOBILE DEVICES

Copyright © 2023 by Michael Bazzell

First Published: February 2023

Project Editors: Anonymous Editor #1, Anonymous Editor #2

Cover Concept: Anonymous Podcast Listener

All rights reserved. No part of this book may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without permission in writing from the author.

The information in this book is distributed on an "As Is" basis, without warranty. The author has taken great care in preparation of this book, but assumes no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Rather than use a trademark symbol with every occurrence of a trademarked name, this book uses the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

Due to the use of quotation marks to identify specific text to be used as search queries and data entry, the author has chosen to display the British rule of punctuation outside of quotes. This ensures that the quoted content is accurate for replication. To maintain consistency, this format is continued throughout the entire book.

The technology referenced in this book was edited and verified by a professional team for accuracy. Exact tutorials in reference to websites, software, and hardware configurations change rapidly. All tutorials in this book were confirmed accurate as of February 1, 2023. Readers may find slight discrepancies within the methods as technology changes.

Revision: 2023.06.18

# CONTENTS

PREFACE

INTRODUCTION

CHAPTER 1: Device Selection

CHAPTER 2: OS Installation

CHAPTER 3: Device Configuration

CHAPTER 4: DNS Configuration

CHAPTER 5: Push Services

CHAPTER 6: Application Installation

CHAPTER 7: Cellular Service

CHAPTER 8: VoIP Service

CHAPTER 9: Data Service

CHAPTER 10: Secure Communications

CHAPTER 11: VPN Configuration

CHAPTER 12: Device Customization

CHAPTER 13: Maintenance & Troubleshooting

CHAPTER 14: Daily Usage & Best Practices

CHAPTER 15: Reset and Reversal

CHAPTER 16: Apple iOS Considerations

CONCLUSION

These contents are provided as a summary. Page numbers and hyperlinks are not included because this is a living document which receives constant updates. Please use the search feature of your PDF readers to find any exact terms or phrases, as that is much more reliable than any index.

# ABOUT THE AUTHOR

## MICHAEL BAZZELL

Michael Bazzell investigated computer crimes on behalf of the government for over 20 years. During the majority of that time, he was assigned to the FBI's Cyber Crimes Task Force where he focused on various online investigations and Open Source Intelligence (OSINT) collection. As an investigator and sworn federal officer through the U.S. Marshals Service, he was involved in numerous major criminal investigations including online child solicitation, child abduction, kidnapping, cold-case homicide, terrorist threats, and advanced computer intrusions. He has trained thousands of individuals in the use of his investigative techniques and privacy control strategies.

After leaving government work, he served as the technical advisor for the first season of the television hacker drama *Mr. Robot*. His books *OSINT Techniques* and *Extreme Privacy* are used by several government and private organizations as training manuals for intelligence gathering and privacy hardening. He now hosts the *Privacy, Security, and OSINT Show*, and assists individual clients in achieving ultimate privacy, both proactively and as a response to an undesired situation. More details about his services can be found at [IntelTechniques.com](http://IntelTechniques.com).

# MOBILE DEVICES PREFACE

I wrote my first privacy-related book in 2012 titled *Hiding From The Internet*. This eventually evolved into the title of *Extreme Privacy*, which is now a large 517-page textbook in its fourth edition, released in early 2022. In early 2023, I began conversations with my staff about the potential for a future fifth edition. There was some resistance. We had just released the 550-page *OSINT Techniques* textbook and we were all exhausted from the process. The idea of attacking a new version of *Extreme Privacy* seemed too overwhelming at the time. We began throwing around the idea of a smaller book.

Many readers of *Extreme Privacy* expressed frustration at the overall amount of information presented within one volume. At 320,000 words, it could be overwhelming to digest all at once. Other criticism was that readers did not necessarily need all of the information within the book. Some wanted to focus on trusts, LLCs, and nomad domicile, and did not need all of the technology-themed chapters. Others only wanted to learn about secure computers, mobile devices, and other technical topics, and did not care about my ideas on an anonymous home or car. This was helpful feedback, and impacted the decision to release this this digital book.

The most criticism from *Extreme Privacy* was about the format. My large OSINT and Privacy books are only available in print. This has upset many readers who want to avoid Amazon or prefer to read on a screen. With this release, we are only providing a PDF. There are no official print versions and we have eliminated Amazon from the entire publication process. This allows us to offer a lower price, and 90% of each purchase directly supports our podcast. If you bought this, thank you for your support!

We realize that a native PDF will lead to immediate piracy of this work online. We accept that. We believe that we can offer further benefits to legitimate purchasers by offering free updates when appropriate. If we ever need to modify existing content or add entire new sections, we can send an email blast to all purchasers which will allow them to download a new copy with all updates for free. Since each copy of this work is watermarked with both a visible and hidden unique code, we can block updates from those who publish the book without consent. Overall, we want to reward those who support us with a searchable, copyable, updatable, and printable document, even at the risk of losing half of our sales to the pirates.

With *Extreme Privacy: Mobile Devices*, I present a new approach to our tutorials. It is not a replacement for *Extreme Privacy* (the printed book). Please consider it a much more thorough supplement about mobile devices. My hope is that this could become a series of shorter volumes which each focus on one specific topic. This would allow people to only purchase the content which they need at a more affordable price. I think it would also allow us to publish content more rapidly, as we can focus our efforts on one area at a time as needs arise, instead of reserving content for a potential future release. Free updates are also a win for everyone. Time will tell if this works. For now, I present this new PDF solely about building the optimal mobile device.

# INTRODUCTION

I believe the most vital step toward obtaining an advanced level of privacy and security is replacing your mobile device and cellular account. Some privacy enthusiasts will tell you that you cannot possess a cellular telephone and still expect any privacy. They have a point, but that is unrealistic. If I informed my clients during an initial meeting that they could never use a mobile app again or send a text message while on the run, I would have no more business. My goal is to allow you to enjoy the benefits of technology while providing minimal data to the companies that benefit most from your usage.

Think about mobile devices from a privacy perspective. We purchase our phones with our own money, pay a monthly fee for cellular connectivity, and carry it everywhere we go. We create and log in to a Google or Apple account for convenience without considering the many ways our data will be abused. The devices are in constant communication with various cellular towers, and their precise location is documented within permanent storage. Apple or Google is collecting information about us and our usage every minute, and then sending the data to their servers for their own analysis and benefit. Our cellular providers allow the software on our phone to pass location data to third parties. Who cares? I do, and so should you. Consider the following.

Any court order demanding your full activity will immediately disclose your location history and all communications. A log of your phone calls and text messages are archived forever and could display an interesting story based on your communication patterns. Your location history could identify your home address, the other places you spend the night, and people you visit. It might identify your habit of speeding down the interstate or identify the organizations with which you hold a membership. However, it gets worse.

A geo fence warrant which has no association with you could disclose your location details even though you were never a suspect. These broad demands provide a dump of data to investigators which identifies any device within an area where a crime occurred. The information is then permanent record within the investigating agency and prone to leaks or FOIA requests. In 2018, Jorge Molina was arrested and held in jail for six days for murder. After receiving a search warrant, Google provided a list of all mobile devices which possessed Google accounts and were located within the area during the crime. Molina's device was on it; however, he was not the killer. He was never even near the crime. He was released after police identified the true killer, but the damage was done. He still has a public record of being arrested for murder.

An employee of Apple, Google, or your cellular provider could also access all of this information with ease. We have seen numerous incidents where employees do bad things for personal gain or revenge. I trust no one.

Finally, we must consider the potential for a breach. As I write this, T-Mobile accidentally leaked 37 million customer records including full names, phone numbers,

home addresses, email addresses, and dates of birth of subscribers. Numerous previous breaches and leaks are already publicly available. These databases are then traded, sold, and abused by strangers. They are devoured by marketing agencies.

What can be done? I believe we can remove ourselves from these risks. This book will help you create a device which does not send data to Apple or Google. Cellular service will be obtained in an alias name, and it will be affordable. A Google or Apple account will not be required in order to download applications and have full-functionality of the device. A true name and physical address will never be associated with the device or service. You will possess numerous numbers within one device which will allow you to protect your true cellular number from the threats previously explained. There will be no more concerns about SIM swapping or account takeover. We will all take our privacy back.

This entire book is designed for the reader interested in extreme privacy. At times, I will assume that your physical safety is in jeopardy, and that making any mistake is life or death for you. I will treat you like a client who is running from a homicidal former lover that is determined to kill you. I will assume that your adversary is tech-savvy. I will never consider costs of products or services, as your privacy and security are more valuable.

I will not sugar coat my opinions or offer less-secure options for the sake of convenience. I will explain every step and will never make assumptions on the reader's level of technology awareness. This is our entire playbook for every new client's mobile device. It is comprised of our internal client tutorials and staff handbooks, with extended details provided by myself. It should allow you to create a perfect private and secure mobile device for your needs. I leave nothing out, and include many new strategies previously omitted from *Extreme Privacy, 4th Edition*.

I offer one last vital piece of information before we start. I encourage you to generate your own opinions as you read along. You may disagree with me at times, which is ideal. That means you are really thinking about how all of this applies to you. **If everyone unconditionally agrees with every word I say, then I am probably not saying anything interesting. If this book only presented content which no one could dispute, then there was no need for the text.** Please read with an open mind and willingness to try new things. Let's begin.



# CHAPTER ONE

## DEVICE SELECTION

I should present the bad news now. If you want extreme privacy, you need a new mobile device. Clients often ask me if they can simply factory reset their current phone, and my answer is always no. Consider the following arguments.

Assume that you are a hardcore Apple user. You have a MacBook laptop and an iPhone device. Every Apple product possesses an embedded serial number. This number is associated with your Apple account. Both mobile and laptop devices constantly communicate with Apple servers, supplying the identifiers associated with your devices. Hard resetting (wiping) an iPhone does not reset the serial number.

Apple still knows who you are. Creating a new Apple ID for use on these devices does not help. Apple maintains a log of all Apple accounts connected to any device. A court order to Apple, a rogue employee, or a data breach can immediately associate your new account to your old, and all of your accounts to all of your hardware. This includes location data and IP addresses. There is simply no way around this. Apple requires an AppleID account to download free apps to your device. Sneaky.

This also applies to most Microsoft and Google products. If you have a stock Android device, Google collects unique identifiers from the device and attaches them to your account. They also store any telephone numbers associated with the device along with unique identifiers within the modem. Since Google also requires an online account to download from their Play Store, they get to collect information about your usage of their email, voice, photos, YouTube, and other services. Wiping the device and attaching new cellular service and a new Google account will fool no one who has the authority to take a peek.

Therefore, we obtain new equipment. It is time to replace your mobile device. For my clients, I arrive with the new equipment in order to ensure it is not associated to them at the time of purchase. Whenever possible, I pay with cash at an electronics store, provide no personal details, and walk out with clean equipment. My image (barely visible under my cowboy hat) is stored on their surveillance system for years, but is not the client's presence. If you plan to buy new hardware with cash, you may want to find a nominee that does not care about privacy to go in the store and make the purchase on your behalf. This is a bit extreme, but justified by some.

During a phone call to an Apple store on my podcast, a manager admitted that every store's surveillance footage is routed to a central collecting location, and stored for an undetermined time. I assume forever. I also assume facial recognition is applied or will be implemented in the future.

Some advocate for buying used devices in order to further confuse the systems that collect user data. I do not endorse this. You never know what you are buying. What if the previous owner was a drug kingpin being monitored by the DEA? A court order

to Apple shows the DEA agent that the device is now being used by a new account. They would have the legal authority to secretly monitor you.

While that would be a very rare occurrence, the possibility of purchasing stolen equipment is much more feasible. If the police show up at your door because your cellular carrier provided the current location of a stolen phone, you will be required to identify yourself. Your name and home address will be included in a report, which is public information with a simple FOIA request. You will be able to explain the purchase, but the damage will be done. All of your hard work at anonymity will be ruined.

The most likely negative outcome from purchasing used equipment is a locked device. It could be stuck within a contract through a specific carrier and you will not be able to activate any service until that debt is paid. If this sounds impossible to you, read some negative reviews on Swappa. You will find countless people who purchased a useless locked device because they wanted to save a few bucks.

We can prevent these situations by purchasing new equipment from retail stores. The minimal extra cost now provides peace of mind while continuing your privacy journey later. I never purchase devices online because there is an immediate permanent digital trail. Even if I used an alias name for the transaction, the device was delivered somewhere and purchased with a credit or debit card which is attached to a bank account. The seller has documentation of unique identifiers for the device. All of this can be tracked. **Cash at a BestBuy or other store is much more private.** Fortunately, the devices we will be using are plentiful in retail locations.

We should probably have the Apple vs. Google discussion now. There are hardcore Android users reading this who never want to use an Apple product. They refuse to pay the "Apple Tax" by switching over to an overpriced ecosystem. They want control of their devices and the ability to make modifications which Apple would never allow. There are also hardcore Apple users who prefer the shiny visual pleasantries of iOS and would never lower themselves to an Android device. They love the convenience of transitioning an Apple account to a new device every year with very little effort. The data magically shows up every time. I understand the cravings of both sides, and I believe either can be satisfied by the end of this book.

I am not an Apple fanboy, but **I do believe the iOS operating system and hardware on the Apple platform is more secure and private than any official default STOCK release by Google (Android).** I do not like the constant data transmissions that Apple collects and stores about your device and usage, which I believe is just as bad as the data collection and usage from stock Google products. Fortunately, we can avoid all data collection by both Apple and Google with a custom phone which is explained in a moment.

In previous years, I pushed Apple iPhone devices onto my clients since they were the best easily available option. Most clients were most familiar with iOS anyway, and very few were willing to adopt something new. Since then, we have witnessed Apple continuously add new data-collection features in effort to enhance the overall iPhone

experience. Today, the only phones I provide directly to clients are custom Android devices which are both private and secure.

I no longer carry any iPhone or other iOS device and I insist my high-target clients do the same. I would also never consider a stock Android device. The amount of personal data forced to be shared with Apple and Google is too much, even with an "anonymous" user account. Instead, **I combine reliable Android hardware with un-Google'd Android software to create our best option for privacy and security.**

After I present these new optimal mobile device strategies, I offer my previous methods of using Apple devices as privately and securely as possible. However, I ask you to read through the entire book before continuing with your Apple device. I believe you will agree that removing yourself from these invasive companies is worth the slight hassle. If you still want to proceed with an iPhone, many of the strategies within the following chapters will still apply, especially regarding DNS, VPN, VoIP, and other technologies.

In *Extreme Privacy, 4th Edition*, I presented four Android paths for consideration. I encouraged readers to consider GrapheneOS as their mobile device operating system, but also explained other options. I walked the reader through custom ROMs, such as LineageOS, since they supported a larger number of devices. I also explained how one could use Terminal commands to modify a stock Android system and disable undesired applications. I even offered an example of building your own Android Open Source Project (AOSP) build and flashing it to a supported device. In this book, I only present one Android consideration: GrapheneOS.

This decision will trigger some readers. There are loyal fans of various secure Android systems such as LineageOS, CalyxOS, /e/OS, CopperheadOS, and others. I have great respect for any community which contributes their work toward our privacy and security. However, I only recommend GrapheneOS to my clients, and it is the operating system I use every day. I also want to eliminate unnecessary complexity of choice by presenting every possible option. However, much like the previous Apple disclosure, much of this book can still be applied to other custom Android-based operating systems.

I believe GrapheneOS is the ultimate solution for our needs. It is the only option which meets all of my requirements, including the following.

- It is completely open-source software which converts a traditional Google Pixel device into a pure native Android environment, including many enhanced privacy and security features, without any Google network services or connections.
- It has a large community testing and verifying any changes, and updates are much more frequent than other builds.
- It provides only the basics and allows you to customize the software you need.
- It has a locked bootloader and does not require root access.

- It allows sandboxed Google push services if appropriate for your needs which can easily be disabled or removed completely if desired.
- It does not require microG for notifications.

All of this, and much more, will be explained later. I carry a GrapheneOS Pixel device with me every day for all communications. It is also my only travel and home device (much more on this later). However, there is no elitism here. Make the best decisions for your own situation. You may prefer another option. Most of this book will apply to any custom un-Googled ROM, but I will only reference GrapheneOS throughout. Take your time, understand the techniques, and make educated decisions about your own mobile device usage.

Much of this book will appear very technical, but the final product we create will possess more privacy, security, and anonymity than anything you can buy off a shelf. I assure you that anyone is capable of completing this process, regardless of your understanding of the technology. I will explain everything, somewhat painfully at times, to make sure no detail is omitted.

GrapheneOS eliminates all data collection by Google, and introduces "Full Verified Boot" within a minimalistic custom operating system. Verified Boot ensures all executed code comes from a trusted source, such as GrapheneOS. It establishes a full chain of trust from the hardware to the software. During the boot process, each stage is verified for authenticity before data can be accessed. It basically makes sure no one has tampered with the system.

Typically, uploading a custom OS to an Android device requires you to unlock and disable this bootloader. After the operating system is installed, the bootloader must remain unlocked in order to use this unofficial build. The unlocked bootloader presents a vulnerability. If I physically took your device; uploaded my own malicious software to it; and then put the phone back, you may not be able to tell. Your data and apps might all look the same, but I could monitor your usage if I modified the OS to do so. This may seem far-fetched until it happens to you.

This is where GrapheneOS has an advantage. After installation, you re-lock the bootloader for additional protection. It then detects modifications to any of the operating system partitions and prevents reading of any changed or corrupted data. This protects the device from many attacks. The authenticity and integrity of the operating system is again verified upon each boot. I cannot unlock the bootloader without deleting all personal data encrypted within the device. Your data is safe.

Because of this, a Google Pixel device is required to install GrapheneOS. Some may be surprised at that sentence. Yes, I recommend a Google Pixel device. This is because we will completely remove all software included with the device and replace it with better versions. Pixel devices offer superior hardware security capabilities than most Android devices, and a Pixel is required for GrapheneOS. Which device should you purchase? That is a personal choice, but understand your options. At the time of this writing (January 2023), the following Pixel devices were supported by GrapheneOS.

Pixel 7 Pro  
 Pixel 7  
 Pixel 6a  
 Pixel 6 Pro  
 Pixel 6  
 Pixel 5a  
 Pixel 5  
 Pixel 4a (5G)  
 Pixel 4a

Any of these devices could be purchased today and possess GrapheneOS within an hour. However, these are not all ideal options. The 4a, 4a (5G), 5, and 5a will all stop receiving support from Google by the end of 2023. This means that they will also likely stop receiving updates from GrapheneOS. If you possess one of these devices and want to use it for testing, or as a personal device until late 2023, I have no objection. If you are purchasing a new device for long-term use, I highly recommend a 6, 6 Pro, 6a, 7, or 7 Pro. If you are reading this in the summer of 2023, the 7a is probably available. In late 2023, we should see the 8 and 8 Pro's appear. Which will you choose? Maybe I can help.

The "a" versions are considered the affordable options for most users. They are very similar to the flagship releases, but are often slightly limited in features. As an example, the 6 and 6 Pro have better and larger displays; possess more RAM; and include nicer cameras than the 6a. They are also twice the price. If you want a premium camera and top speed, then those flagship models may be appropriate. However, the 6a is much more than sufficient for our needs, and is quite affordable. I purchased a Google Pixel 6a specifically for writing this book for \$300, paid in cash at a local BestBuy store during a holiday promotion. Today they can be found for \$399.

I personally carry a Pixel 6a and it is the default option I provide for my clients. When the 7a's arrive, I will very likely transition to the updated model. All 6, 7, and future Pixel devices will receive security updates for five years. This also translates to the likelihood of five years of GrapheneOS weekly updates. My 6a should be supported until July 2027.

Much of my desire for the "a" model is size. They are typically smaller than the flagship options. I prefer a smaller device. I think the 6a is still too large and crave the days when my 4a was top of the line. If the upcoming Pixel 8 possesses a smaller footprint, as rumored to be true, then I would consider that over the "a" series. For now, my 6a meets all of my needs.

Once you have identified the appropriate model for your usage, please only consider "unlocked" devices. Some stores will push you toward a device which is designated for a specific carrier such as T-Mobile or Verizon. While there may be a slight financial incentive for this restricted device, it will not work for every tactic presented within this book. By purchasing an unlocked device, you have the freedom to choose your cellular service provider at any time. Later, I present a strategy which allows you to connect to any provider's towers for service, and an unlocked device will be crucial.

I would like to remind readers that every mobile device will be replaced with the latest and greatest at some point. The 6a is much more powerful than devices from only a year prior. Please don't try to constantly chase the fastest and best thing out there. You might drive yourself crazy. The "budget" phone of today is usually better than the flagship of yesterday.

Consider your needs. Purchasing the most expensive device will probably result in wasted processor limits and unused RAM. Unless you constantly play the latest games, take professional photos, or only watch 4K movies on your device, you do not need anything expensive. If you need all of that, you probably will not like our final private and secure device anyway. Most readers in early 2023 will find the 6a to be their best option. Once the 7a is available, consider that device.

# CHAPTER TWO

## OS INSTALLATION

Once you have obtained the best device for your needs, you are ready to install GrapheneOS. There are two options for installation of GrapheneOS onto your Pixel device. The web installer is the easiest for most users, while the Linux method is most stable for those without a chromium-based browser. I will discuss both. However, the web installer should work for your needs.

### Prepare the Device

Regardless of the installation path you choose, you must first prepare the phone itself. Turn on the Pixel device and dismiss any attempts to enter a Google account. On my new unit, I had to conduct the following.

- Click "Get Started", "Skip", then "Set up offline".
- Click "Continue" then "Next".
- Deselect all options and click "Accept" then "I Accept".
- Click "Skip", confirm "Skip", and "Skip" again.

Swipe the menu up to find and launch "Settings", then navigate to "System Update" and apply all pending updates. Reboot and continue to apply updates until none are available. Note that your device will require internet access via Wi-Fi to complete the process. This could take some time, especially if this is a brand-new device with Android 12. It is vital to patch the phone to the latest Android build before we proceed. When all updates are applied, conduct the following.

- Navigate to "System Update" and apply all pending updates.
- Tap "About phone".
- Tap "Build number" several times until "Developer mode" is enabled.
- Tap the back arrow then tap "System".
- Tap "Developer Options".
- Enable "OEM Unlocking" and "USB debugging".

If "OEM Unlocking" is still greyed out and unavailable, you must conduct a full factory reset before you proceed. I conducted the following. Skip this section if you completed the previous steps.

- Remove any fingerprints from "Settings" > "Security" if applicable.
- Remove any accounts from "Settings" > "Passwords & accounts" if applicable.
- Click "Settings" > "System" > "Reset options" > "Erase all data".
- Reboot, enter system, connect Wi-Fi, and wait 2 minutes.
- Enable "OEM Unlocking" and "USB debugging" using the previous tutorial.



## Browser-Based Installation

We can now install GrapheneOS. I will begin with Web Installer. From your Windows, macOS, or Linux computer, make sure you have a Chromium-based browser installed. If you have Chrome available, it should work fine. If you do not have Chrome, which I do not due to Google's privacy invasions, download and install Brave Browser (brave.com). This provides the stability of Chromium but lacks most of the invasive software included with Chrome. You can uninstall it when finished if desired (I did).

Next, navigate to **<https://grapheneos.org/install/web>** and read through the entire page. Once you understand the overall installation process, run through the steps, which are outlined next. **Always rely on the official GrapheneOS page for any changes since publication.** The following are the steps required at the time of writing this chapter. Make sure you have only one browser open and only one browser tab available.

- Turn the device off.
- Hold the power and volume down buttons simultaneously.
- When you see the "Bootloader" menu, release the buttons.
- Connect the device to computer via USB cable.
- Click the "Unlock Bootloader" button on the GrapheneOS page.
- Select your device from the popup menu.
- Click "Connect".
- Press the volume down button on the device to change options and highlight "Unlock Bootloader".
- Press the power button to confirm the choice.
- Click the "Download Release" button on the GrapheneOS page.
- Allow the appropriate version of GrapheneOS to completely download.
- Click the "Flash Release" button.
- Allow the process to complete.
- Click "Lock Bootloader" on the GrapheneOS page.
- Press the volume button on the device to select "Lock Bootloader".
- Press the power button to confirm the choice.
- Make sure "Start" appears next to the power button and press it.
- Allow the phone to boot.

This sounds simple, but a lot can go wrong. In my experience, only Chrome-based browsers will reliably complete the process, but the choice of operating system itself should have no impact. Chrome, Chromium, and Brave browsers within Windows, macOS, and Linux should all work the same. Attempts with Safari and Firefox failed for me. A poor-quality USB cable can also ruin the entire process, so use the cable included with the device when possible. Some Windows machines may not have the appropriate drivers for your device. If the phone is not recognized, plug it in and attempt a software update at "Windows Update" > "Check for updates" > "View Optional Updates". If you now have GrapheneOS installed, skip past this next section about installation through Linux to continue.



## Linux-Based Installation

Before we proceed, I want to issue a warning about the following process. You must be absolutely sure that you have replaced my demonstration commands with the current commands appropriate for your exact model and current version of GrapheneOS. If you were to replicate my commands on a different model of Pixel, you might "brick" the device and it could be worthless. It may never boot again. The previous web-based method automatically detects the model of your hardware and installs the most current stable version of GrapheneOS. I discourage most users from the following Linux-based installation. If you insist on installation via Linux, please continue.

The following steps were slightly modified from the GrapheneOS website at <https://grapheneos.org/install>. **Always check that site before proceeding as things may have changed since this writing.**

The following tutorial requires an Ubuntu Linux computer, and I used a laptop with Ubuntu 22.04 as the host. This is the cleanest and easiest option. While you can install from a Windows or Mac host, software requirements can vary and driver issues can be complicated. The Linux steps are more universal. Never use a virtual machine for this installation due to USB detection issues.

We must now configure software within our Linux computer. Conduct the following within an Ubuntu Terminal session. Note that the exact version presented here may have been updated since this publication was released. The tutorial steps offered at <https://grapheneos.org/install/cli> will be updated as needed. These steps also install ADB, which is required within other tutorials.

- `sudo apt install libarchive-tools`
- `curl -O https://dl.google.com/android/repository/platform-tools_r33.0.3-linux.zip`
- `echo 'ab885c20f1a9cb528eb145b9208f53540efa3d26258ac3ce4363570a0846f8f7 platform-tools_r33.0.3-linux.zip' | sha256sum -c`
- `bsdtar xvf platform-tools_r33.0.3-linux.zip`
- `export PATH="$PWD/platform-tools:$PATH"`
- `sudo apt install android-sdk-platform-tools-common`
- `fastboot --version`

The final command verifies that Fastboot is installed which should display the version number. We now need to boot our device into the bootloader interface. To do this, hold the power and volume down buttons simultaneously while the device is off. This should present a "Fastboot mode" menu. Connect the device to your Ubuntu computer via USB cable. Execute the following command within Terminal and verify it displays "OKAY".

- `fastboot flashing unlock`

Press the volume down button on the mobile device until "Unlock the bootloader" is displayed, then press the power button. We are ready to download the new operating system files. First, you must navigate to [grapheneos.org/releases](https://grapheneos.org/releases) and select your device within the "Stable Channels" section. Note that the 6a is code-named "bluejay", while other models are code-named "oriole" (6), "raven" (6 Pro), "panther" (7), and "cheetah" (7 Pro).

Next, identify the latest version number, such as "2023012500". You will need to replace each version within the following examples (2023012500) with the latest version displayed on the website during your installation. **It is vital to confirm all of these steps at the official GrapheneOS website and to choose the correct version for your device!** Always double-check that you are entering commands for your specific model. Execute the following within Terminal **ONLY** for the Pixel 6a (bluejay).

- `sudo apt install signify-openbsd`
- `alias signify=signify-openbsd`
- `curl -O https://releases.grapheneos.org/factory.pub`
- `curl -O https://releases.grapheneos.org/bluejay-factory-2023012500.zip`
- `curl -O https://releases.grapheneos.org/bluejay-factory-2023012500.zip.sig`
- `signify -Cqp factory.pub -x bluejay-factory-2023012500.zip.sig && echo verified`

The last command should display a confirmation that the software is correct. This confirms that we have downloaded a secure file which has not been intercepted or maliciously replaced. The following Terminal steps extract the download and install it to the device.

- `bsdtar xvf bluejay-factory-2023012500.zip`
- `cd bluejay-factory-2023012500`
- `./flash-all.sh`
- `fastboot flashing lock`

You should now see the option "Do not lock the bootloader" on the device. Press the volume down button until "Lock the bootloader" is displayed and press the power button. You can now reboot the device by pressing the power button labeled "Start" or holding down the power button to turn off, and then turning on as normal. Allow the phone to boot without making any selection.

## Device Boot

Once GrapheneOS is installed, you are ready to boot it for the first time. You should immediately see a warning of "Your device is loading a different operating system". This is completely normal, and is Google's way of trying to lure you back to their invasive system. This is safe to ignore. Upon first boot of GrapheneOS, press "Start" and "Next" until the Wi-Fi connection screen is present. Connect to Wi-Fi and complete the following tasks, with considerations for each.

- Click "Next" if prompted about the SIM card missing.
- Disable location services for now, this can be set up later if needed.
- Skip the fingerprint setup for now.
- Assign a secure PIN for the screen lock.
- Skip any restore options.
- Click "Start".

Your installation is now complete. The device itself is completely encrypted and sends no data to Google. Next, let's harden a few settings. Once you are within the new operating system, confirm that OEM unlocking and developer options are disabled with the following steps. This may be redundant, but we want to make sure we are protected.

- Swipe the menu up to launch "Settings" and click "About phone".
- Tap "Build number" at the bottom until "Developer mode" is enabled.
- Enter your PIN if required.
- Click the back arrow and click "System" then "Developer options".
- Disable "OEM Unlocking" and confirm the choice.
- Disable "Developer options" and reboot the device.

I believe GrapheneOS is not only the most private and secure mobile device option we have, but it is the most elegant and minimalistic. It has no bloatware or undesired apps. I must admit that half of my clients do not use GrapheneOS and still prefer iOS. Only those with extreme situations have successfully made the switch. I trust that you are now ready to fully configure your optimal private and secure mobile device.

# CHAPTER THREE

## DEVICE CONFIGURATION

Your new GrapheneOS device is now very private and secure by default. Most of the settings are optimally configured and the device is ready to use. However, I believe there are some adjustments which are beneficial to readers. Additionally, it is important to understand the default customizations created by GrapheneOS. This operating system is not simply a new skin of Android which is missing Google services. Every facet of the system has been tweaked for the sake of privacy and security. While beneficial to most, this could cause hiccups in your daily usage if you are unaware of these changes. Therefore, let's walk through everything so you can make the best decisions for your device. By the end, I hope you see the many benefits of this operating system and are convinced to leave Apple and Google behind.

Please note that this was written in January of 2023 with the current version of GrapheneOS. By the time you read this, some of these settings may have changed or disappeared. When that happens, this PDF will be updated and purchasers will be notified to download a new free copy. Please use this guide as an overall explanation about a typical configuration of GrapheneOS or other custom builds. If something has changed, research the new options and proceed. If you are reading this in 2030, we may not even have physical mobile devices any more.

I typically start at the top of the home screen, and then work my way through the settings. If you swipe down from the top, you will see the stock "Quick Settings" menu. Swiping down a second time will switch the view from compact into full. The pencil icon in the lower right allows you to edit this menu. The figure on the following page (left) displays the default full menu after installation.

Once in the edit menu, you can tap, hold, and drag options down to remove them, and tap, hold, and drag options up to add them. Everyone's preference is different, but I will explain my desired layout. I prefer the top two buttons to allow enabling and disabling of internet and airplane mode. This allows me to quickly turn off Wi-Fi and cellular connectivity, or connect to either when I need access. This is fairly basic and common functionality.

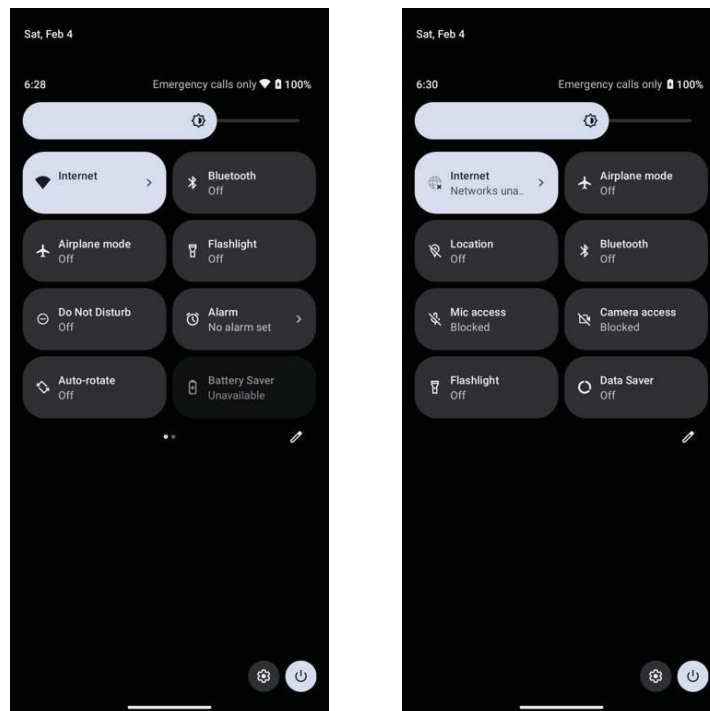
I prefer the second row of buttons to allow enabling and disabling of location and Bluetooth. I rarely enable (allow) these options, and their presence is more of a comfort knowing that neither are active. Again, this is common behavior. My third row of buttons is where things get interesting. The "Mic access" and "Camera access" buttons offer a convenient way to enable or disable all microphones and cameras with one click. Surprisingly, not all stock Android 13 devices offer this option.

I typically leave these both disabled (blocked) at all times. If an application wants to access either a camera or microphone, you will be prompted to allow this activity. As an example, consider an incoming video call over Signal. If both options are disabled, you will receive two separate popup menus when the call comes in. The first should

ask you if you want to enable your microphone, and the second will confirm you want to enable the front-facing camera. Declining these will continue to the call, but the other person will not be able to see or hear you.

This is a great feature, and one that I rely on daily. It prevents accidental sharing of audio or video, but it is not perfect. This is still software-based blocking. It is easier to accidentally allow transmission due to an unintentional button click than to remove camera cover stickers or a physical microphone blocker plugged into the USB-C port. However, these features provide great protection when properly used. Note that these options are not re-disabled after a call. You must manually return to the quick menu and tap each again to continue blocking audio and video.

Finally, I add the flashlight and data saver buttons to my last row for easy access and remove any others. The data saver will become vital once I explain optional international data-only eSIMs which allow access to any network, but are prepaid per megabyte (MB). This feature limits the data being used in the background to minimize our bill. The following figure (right) displays my final quick menu.



The default GrapheneOS home screen is fairly minimal, and I make no modifications at this time. Later, we will take advantage of a custom launcher which will provide a better visual presence. For now, we can swipe up from the bottom to see the default applications. This application drawer should appear quite minimal compared to traditional Android devices. Notice there are no undesired social network apps, forced Google services, or streaming video trials which are impossible to uninstall. We possess only the basics, which is how it should be. It is now time to enter the "Settings" application and begin exploring our new options. I will not visit everything present here, but I will highlight areas of interest.

I prefer to completely configure devices before they ever touch a cellular network. At this point, I navigate to "Settings" > "Network & internet" and connect via Wi-Fi. I conduct this on my home network while behind a VPN-enabled home firewall, but that is not mandatory. Remember that GrapheneOS is not sharing data with Google, and not tracking your activity. If you are a follower of my extreme tactics within my books, you may want to connect behind a firewall or public Wi-Fi. Connecting direct to your home internet is acceptable for some, but know that you are sharing your home IP address with every service and application you install. See *Extreme Privacy, 4th Edition* for more details.

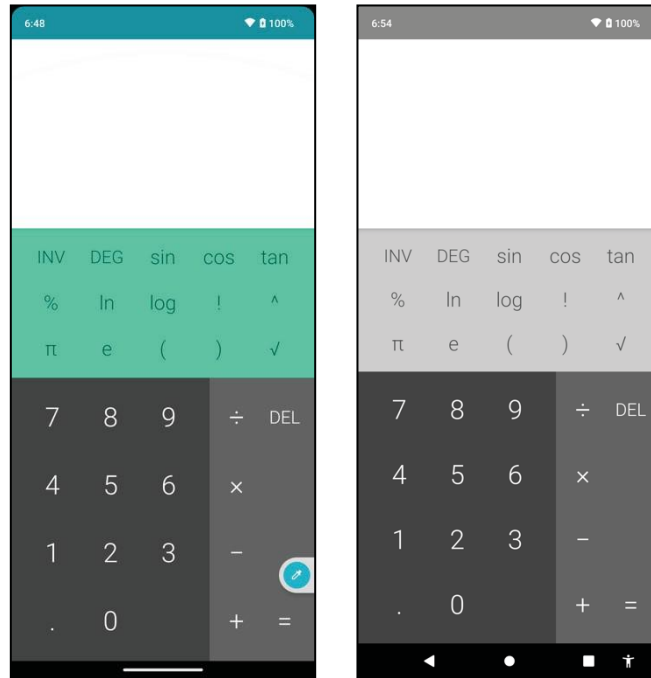
Within the "Internet connection" menu, you will see an option called "Network preferences" near the bottom. Clicking this provides options which may seem new to you. The default setting within GrapheneOS for "Turn off Wi-Fi automatically" is set to "disabled". Most stock devices do not offer this option. If you have Wi-Fi connected to your stock Android or iPhone device, and leave the area, the Wi-Fi service stays active on your device. It is constantly looking for any known networks while broadcasting unique identifiers from your device to any sniffing hardware which may want to track you. When you return to a known network, it automatically reconnects.

I change this to "1 minute". With this new setting, your device will disable Wi-Fi altogether one minute after being outside of the connection. This is a great feature and automatically disables my device's Wi-Fi when I forget to disconnect. This prevents my device from broadcasting Wi-Fi details while I am out. I then make sure "Turn on Wi-Fi automatically" and "Notify for public networks" are each disabled.

Next, go to the "Settings" > "Security" menu. If desired, add a fingerprint to the system for easy unlocking. This does not save your actual fingerprint to the device, but it creates a series of calculations to know if the correct fingerprint is being used. Most of my clients apply this. I then disable the "Native code debugging" and "Allow camera access" options. In the "Settings" > "Safety & emergency" > "Wireless emergency alerts" menu, I disable everything. I find these to be an annoyance and rarely relative to my current area.

I then navigate to "Settings" > "Accessibility" > "Color and motion" > "Color Correction" and enable "Use color correction". I then select "Grayscale" and enable the "Color correction shortcut". This places a small floating shortcut on the device which allows me to enter or leave monochrome mode at any time. This is a personal preference, as it allows me to focus on email or other messaging in black and white for distraction-free work. This is completely optional.

I then change the "Settings" > "System" > "Gestures" > "System navigation" to "3-button navigation" and "Settings" > "Accessibility" > "Accessibility shortcuts" > "Accessibility button" > "Location" to "Navigation bar". This minimizes the screen impact of the shortcut. The following image (left) displays the calculator without the color correction and the floating shortcut. The image to the right displays it with color correction enabled and the shortcut (person icon) within the task bar.



Next, we must consider web browsers. GrapheneOS includes a custom Chromium-based browser called Vanadium. It is hardened with security-focused settings and sends no data to Google by default. In previous writings, I encouraged readers to consider Firefox Focus as a daily browser, but I no longer agree with that. I believe Vanadium should be the only browser on the mobile device. Consider the following.

- Multiple browsers present the need to update and maintain additional apps.
- Multiple browsers provide a larger attack surface.
- Vanadium provides strong site isolation with each site in a "sandbox".
- Vanadium relies on the hardened WebView implementation.

While I still use Firefox as my daily browser on all desktop systems, I no longer install it within my mobile device. I believe Vanadium is now the superior option for GrapheneOS. However, there are a few changes I make to the application.

- Open Vanadium and scroll down slightly to see the menu.
- Tap the three dots in the upper right and select "Settings".
- Tap "Password Manager" and disable "Save passwords" and "Auto Sign-in".
- Tap the back arrow.
- Tap "Payment methods" and disable everything.
- Tap the back arrow.
- Tap "Addresses and more" and disable everything.
- Tap the back arrow.
- Tap "Privacy and security".
- Enable "Close tabs on exit".
- Tap the back arrow.

Much of this is personal preference and you should always modify the settings as best for your usage. One thing I liked about Firefox Focus was that it erased all activity every time it was closed. Each opening of the app presented a fresh start with no history, cookies, or cache. Vanadium does not offer this. Instead, we should clear out all data on occasion. This can be accomplished by going to "Settings" > "Privacy and security" > "Clear browsing data". Once there, I choose "Advanced" and "All time"; select all items; then tap "Clear data".

If you would like Vanadium to always launch without any pre-visited sites within open tabs, you can enable "Close tabs on exit" from the "Privacy and security" menu. If you want links from external apps to always open in Incognito mode, you can enable this option within the same menu. I enable both. This leaves a lighter footprint within my browsing history.

Next, I open the camera app, swipe down slightly to present the settings menu, and make the following modifications.

- Select "Optimize for" and "Quality".
- Click "More Settings"; enable "Gyroscope"; and disable "Camera Sounds".

I prefer a quiet device which will not collect attention from those around me. Therefore, I navigate to "Settings" > "Sound & vibration" and make the following modifications. These may be inappropriate for those who need to be notified of every incoming call.

- Change "Phone ringtone" to none.
- Change "Default notification sound" to none.
- Disable "Screen locking sound".
- Disable "Charging sounds and vibration".
- Disable "Touch sounds".

The double-lined clock on the lock screen annoys me. I disable this at "Settings" > "Display" > "Lock screen". While I am there, I add text to the lock screen which contains "Reward" followed by an internet telephone number which can be answered or texted without the mobile device. I explain more on this later.

## **Contacts**

We have been spoiled for many years with contacts synchronization services. When you save a name and telephone number within a traditional device, that data is synchronized to an Apple or Google server, and present within your iCloud or Google account. Storing a number in one location makes it visible on any other device which accesses your account. Fortunately, we do not have that issue with GrapheneOS.

I use the term fortunately because I don't want that feature. My contacts are sensitive and I do not want to share them with strangers. I do not want employees of these companies to be able to access this data. Therefore, we must do things manually. I



currently store all of my contacts within the Contacts application on my encrypted Linux laptop. This program is completely offline and allows the export of a VCF file. The stock GrapheneOS contacts application allows import of this type of file. Once a month, I export a new VCF file; copy it to my USB device; insert that device into my Pixel, and import the VCF file into Contacts. That seems like a lot of work, but it is not too bad. This allows me to know my sensitive contacts never touched the cloud. One could also export contacts from Proton Mail, or any other service which supports VCF files, and import that way.

## **PIN Scrambling**

Recent updates to GrapheneOS have introduced a new security feature which may be desired by some readers. You can now scramble the numbers presented within a PIN entry screen. This could be valuable for people who may fear they are under surveillance and entry of the same pattern on every unlock could provide a way to enter your device when physical access is obtained. Unlocking the device takes longer since the numbers are randomized each time, but there will no longer be an identical pattern present within surveillance video, and the fingerprint smudges on the screen will not be helpful to an attacker. You can enable this feature by navigating to "Settings" > "Security" > "PIN scrambling".

## **Auto Reboot**

GrapheneOS devices will automatically reboot every 72 hours if they have not been unlocked. This is a security feature. In the event your device has been lost, stolen, or seized, and the screen has not been unlocked within three days, it will reboot into a state which would not allow biometrics to be used for access. The PIN would be required. This setting is appropriate for most people. If you have an extreme need, you can modify this setting with the following steps.

- Navigate to "Settings" and then "Security".
- Select "Auto Reboot".
- Modify the setting as desired.

You should now have some strong basic modifications to your own device. Next, we must tackle DNS configurations.

# CHAPTER FOUR

## DNS CONFIGURATION

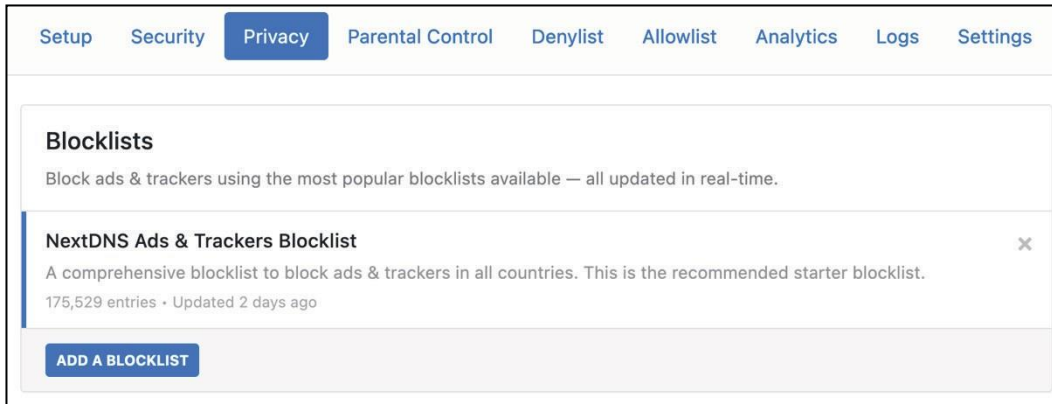
Domain Name System (DNS) can be a very overwhelming topic. In a very basic and simple explanation, DNS translates domain names, such as inteltechniques.com, into IP addresses in order to locate the appropriate content. In a typical setup, your home or mobile internet service provider (ISP) conducts your DNS queries quietly without your input. In other words, your ISP knows every website domain you visit, regardless of SSL encryption, and knows your billing details. If you did not purchase internet or cellular service anonymously, then they also know your identity. ISPs collect a lot of valuable information about you this way, and often sell these details to third parties for marketing purposes. I want to stop that.

By default, your GrapheneOS device relies on the DNS service of the network to which you are connected. This could be your home Wi-Fi or cellular provider's service. You have the option to specify a different DNS server for all queries generated from your device. I implement NextDNS (nextdns.io) service for my devices and the phones of all clients.

NextDNS conducts the DNS queries required in order to navigate your internet traffic, but it also includes filtering options. I will explain with an actual configuration demonstration conducted from my test device in a moment. First, create a new free account at <https://my.nextdns.io/signup>. Any masked or private email service should be accepted and no payment source is required. I used an alias name. The free tier allows 300,000 monthly queries at no cost. After registration, you should be taken to your NextDNS user portal which should display a “DNS-over-TLS” address similar to 12345.dns.nextdns.io. You can now use this address to apply their DNS service and filtering options. Conduct the following on your mobile device.

- Within GrapheneOS, open the "Settings" application.
- Select "Network & internet".
- Scroll to "Private DNS" and tap it to open the options.
- Select "Private DNS provider hostname".
- Enter the “DNS-over-TLS” address provided by NextDNS.

Your Android device is now using NextDNS for DNS queries, and you can see the logs of these requests in your NextDNS portal. This may be alarming to some readers. The "Logs" tab in your portal identifies every connection being made from your device. This can be a privacy concern, but it has many benefits. We can now apply filters which will block many undesired connections. Click the "Privacy" tab and notice the automatically-applied blocklist. If this was not applied, add the "NextDNS Ads & Trackers Blocklist". This database blocks over 100,000 connections which are associated with ads, trackers, and malware. This will block a lot of unwanted connections such as pop-up ads, tracking code, telemetry, and user analytics. You now have greater protection. The following image displays my configuration and the menu.



Note that this feature may block more than you desire. On occasion, my device's browser refuses to display a website I am trying to visit. It is not the device which blocks it, it is NextDNS. If you ever find a site which refuses to load, change your device's DNS option back to "Automatic" and reload the page. If it loads, you know the reason. This is a rarity, but you should understand the solution when needed. I set this list as default for all clients, and never hear about blockage.

Click on the "logs" tab again and take a look at the traffic. Open the Vanadium browser on your device and visit a few websites. Then, refresh the NextDNS Logs page (circle arrows icon) and notice the difference. You will likely see several connections allowed and others being blocked. This is the filter lists in action. If you see a connection being allowed which you do not want to occur, you can copy that domain and add it to the "Denylist" tab. I did this for a domain which was being queried by an application in order to send "anonymous" analytics about my usage. I will provide more detailed examples in a moment.

If you plan to use NextDNS full-time on your device(s), I highly recommend that you modify the logging aspects. Click the "Settings" tab within your NextDNS portal and review the "Logs" section. You can disable logs completely or change the retention period. I choose the latter while I am testing my devices. I leave logs enabled; disable "Log Client IPs"; enable "Log Domains"; and set the retention to "1 Hour". This way, I can always connect to the portal to see what is being blocked and allowed, but the logs will be purged an hour after each activity. I can make modifications while I am configuring my mobile or desktop devices and see my results immediately.

**Once I have all desired NextDNS configurations in place, I disable logging completely.** This eliminates any history of my internet activity through NextDNS. We will rely on these logs in future chapters, so do not disable them completely just yet. Whenever desired, you can purge all logs with the "Clear logs" button.

Once you have NextDNS programmed into your device, refresh the logs in your NextDNS portal. The following displays my result.



This confirms that GrapheneOS conducted a DNS check and confirmed the connection. Notice that this is not Google's DNS verification service, as would have been default on a stock Android device. Next, I navigated to "Settings" > "System" > "System update" > "Check for updates" on my device. I then looked at the logs within NextDNS and noticed a new connection to "releases.grapheneos.org". This confirms that things are working. Since I do not want to block any of these requests, I will take no action. Once we start installing third-party applications, we should see a lot of undesired connections which we can block.

Next, you should ensure that your connections are encrypted. Within Vanadium, navigate to <https://cloudflare.com/ssl/encrypted-sni> and conduct a test. You should see checkmarks next to DNSSEC and TLS. If you do, you are hiding much of your internet traffic from your ISP and your VPN. The other two options on this page apply to Cloudflare's DNS service and can be ignored. For this test, I only care that the traffic is encrypted with a TLS connection.

While you have the Vanadium browser open, visit yahoo.com and allow the entire page to load. If you are familiar with that site, you may notice that the majority of the popup annoyances, embedded videos, and flashing ads are no longer present. This is because NextDNS blocked those connections before they ever reached your device. Next, return to your NextDNS portal and reload the Logs page. It may take a couple of minutes for the results to appear. You should see something similar to the following.



The red bar on the left confirms which incoming connections were blocked. We can see our blocklist in action. On yahoo.com, dozens of ads and trackers were blocked without any effort from us. We do not need any browser extensions or firewall applications. This is the true power of NextDNS. This blocking strategy is much cleaner with minimal resource usage. It also prevents conflicts with VPN applications.

This is all a lot to digest. Let's summarize some of the key takeaways. By default, your internet service provider supplies DNS services, and often uses that data maliciously. When you configure NextDNS on your Android mobile device, you are using their lookup service instead. Furthermore, the blocklists can prevent applications from

sending out telemetry and analytics about your usage. This will become much more apparent in the next chapter.

Remember the limits of the free tier. Most people will not exceed 300,000 queries a month. If you have multiple devices, you can either create an account for each or pay a small fee for NextDNS's premium service.

Some readers may be upset that I have chosen NextDNS over AdGuard as a filtering DNS provider. My reasons are the following.

- I have more trust in NextDNS. The founders are publicly visible and I know who runs the company. They are reputable people who have been heavily involved in this space and are transparent about their reasons for the service.
- A premium-tier business model explains the funding for resources.
- AdGuard is a Russian company which was moved to Cyprus, but their infrastructure remains in Russia. A Russian CEO has minimal presence on the internet, but there is no information about any other owners.
- The support from NextDNS has been superior. When I contacted both companies with questions about the product, only NextDNS responded. One of the owners provided full details.
- AdGuard recently announced a new program similar to NextDNS which will allow custom filtering. However, my emails requesting information were unanswered. The custom options from NextDNS have been thoroughly tested and vetted.

The final privacy consideration in regard to DNS is account-based versus publicly-available servers. While a custom NextDNS account can be wonderful for blocking (or allowing) connections, it does carry some risk. Since you have an account, all queries could be tracked back to a specific user. Disabling logs should prevent this, but a court order could override your configuration. Using an alias name should provide comfort. Public NextDNS servers do not require an account, but provide no custom filtering. If you want to filter ads without an account, I do believe AdGuard is currently your best option ([dns.adguard.com](https://dns.adguard.com)). However, you cannot modify the protection. If they block a domain, there is no way to unblock it. Again, this is where custom filtering from NextDNS is superior.

Are you sick of DNS yet? There are many opinions of the proper way to use DNS services. None of them are perfect for everyone. I hope you take the information presented here and use it as a starting point toward your own DNS and VPN strategy. Please consider the remaining chapters before you lock in your own plan. In Chapter Six, we will rely on NextDNS to further harden our device's privacy and security.

# CHAPTER FIVE

## PUSH SERVICES

Before proceeding to application installation and telephony services, we must have a serious conversation about push services. If you have ever owned a traditional Android or iPhone device, you are familiar with notifications of incoming communications. When an email arrives, you might receive a ding, buzz, or visual notification which can be easily checked. A text message from your favorite messaging app might alert you of a new arrival. This is due to push services. Apple and Google each have their own environment to deliver this data conveniently and with little battery drain.

Since there are no Google services on your new GrapheneOS device, Google is not receiving any data about your usage. While your desired apps should install without issues, everyday function may be a problem. Since GrapheneOS does not contain any Google apps, you are missing some core Google software which provides services such as push notifications, location tracking, and mapping. This may sound like a huge benefit, but it also presents some limitations. You can typically still open apps and "fetch" data such as pending email or text messages at any time, but you might be missing instant notifications.

With some apps, synching of content might be delayed. Some secure messaging apps can deliver messages instantly through their own platform without the need for Google's push service, but at a cost of battery drain. Traditional email applications, such as Proton Mail, may only fetch the data when the app is opened. This may be a desired feature to some. A true Google-free experience without constant incoming notifications is a nice change.

Personally, I prefer to intentionally fetch desired content when needed in order to keep Google or Apple out of my business. My phone never lights up during meetings and never dings audible tones throughout the day. There is never a looming notification reminding me that my inbox is growing with unread messages. I check for any communications on my own time. I am never tempted while driving to check the latest email which just arrived. When appropriate throughout my daily schedule, I check my email and other communications apps by opening each. The content is fetched from the various servers and I can tackle anything which needs a response.

It took a while to lose the anxiety of potential missed messages. Today, it reminds me of the way email was checked when I first started using it. Back then, you logged into your computer; opened your email client; fetched any incoming messages; responded to those desired; and closed the software after the messages had been sent. You then might even turn off the computer and go about the rest of your day. Today, I check my phone often for email and other communications, but it no longer controls my life with instant notifications.

Many readers may think this is an unattainable luxury. I respect that you may have children in school which need to get in touch with you at all times; an employer who

insists you respond to anything within minutes; or a sick family member who needs direct access to you. If you need immediate notification of incoming emails, text messages, or calls without launching applications, that can still be achieved in a minimally-invasive manner, as described in a moment.

Many people discuss installing an open-source version of Google's Push services through software called microG, but that will not work with GrapheneOS. This operating system is hardened very well, and does not allow weakened security through the use of these privacy-leaking options. Before we get too far, let's understand some basic services and determine the benefits and risks of each.

- Default Android devices include numerous Google services which run in the background and assist with communications throughout all of your applications. These services have access at the operating system level and are quite invasive. Google gets to eavesdrop on everything you do. Since most of their code is closed-source, we have no way of knowing what data is being digested and transmitted to Google.
- Custom ROM devices which implement microG replace the closed-source code with open-source alternatives which mimic Google's services. This software still relies on Google's network, and Google still receives a lot of information about your activities, but this is better than the previous option.
- GrapheneOS does not include either Google or microG options enabled by default. If you need Google's services, you must enable a "Sandboxed" version within the GrapheneOS Apps menu. This contains the official Google code, but it is severely restricted, and only has permissions on an application level. This is much different than having full access throughout the entire operating system. Google also only has access to the profile of which it is installed, and a second profile could be created without it (more on that later). I believe this limited version of Google's services is superior to microG.
- Finally, GrapheneOS by default has none of this activated. If you do not need push services and notifications, there is nothing you need to do. If extreme privacy is your goal at any cost, then you should skip to the next chapter. However, that is not realistic for most users.

Before I convince you one way or the other, let's discuss some actual experiences if you do not activate push services. If you use Proton Mail as your secure email provider, you will not receive any notifications of incoming messages. You will need to open the app occasionally and check your email. This would also apply to any other email service relying on Google's network to deliver notifications.

If you use Tutanota as your secure email provider, push notifications work natively within their own network, but the application must remain active in the background. This applies to Signal as well. It has the option to receive immediate notifications of incoming text messages without the need to open the app. Each of these options requires more battery, so you may see faster drain.



If you use Sipnetic or MySudo for telephone calls, as explained later, you will receive notifications of incoming calls only if the application is open and running in the foreground. Your device will ring as normal and you can answer the call. If you are expecting a call and ready to receive it, you should have no issues. If you receive an unexpected call while the device is in your pocket with the application dormant, it may go straight to voicemail. Most other communication applications will not send notifications, and you will need to open those apps in order to see any pending messages.

For some people, the ability to receive incoming calls and secure message notifications through Signal will be sufficient for daily use without the need for any Google services. If your entire family is on Signal, they can reach you. If your child's school will only call a traditional telephone number in the event of an emergency, this may not work for you.

Now, let's compare this to a device with Google's services enabled via GrapheneOS's sandboxed environment. All incoming email, text messages, voice calls, and video calls will ring as they normally would on any other device. The notification bar at top will alert you of all incoming communications and a notification dot can be placed on any app which has pending content. You will not need to peruse through each app to see if someone is trying to get a hold of you.

Which route should you choose? I cannot answer that, but I can offer some assistance by asking a few questions.

- Do you need to have the ability to answer any unexpected traditional telephone calls? If your answer is yes, then you need push services.
- Do you need to be visually or audibly alerted any time an email or text message arrives? If your answer is yes, then you need push services.
- Do you often place your device into "Do not disturb" mode and ignore the barrage of incoming communications? Then you may be fine without push services.

Remember that mobile device privacy is a series of decisions which produce an environment most appropriate for you, and will be unique for everyone. I have a few clients who use GrapheneOS with push services every day and love it. I have others who went without and hated it. It really depends on your personality and desire to be notified of everything at all times. For me, switching to a completely un-Google'd device was therapeutic. It reminded me that I do not need to see everything in real time, and there was life outside of my various networks.

In past writings, I took a strong stance on removing Google 100% from my digital life. I stated very clearly that any hardcore privacy advocate should go without Google's push services. Today, I do not have that same strong resistance. Let's think about what real damage is done if you enable sandboxed push services.



First, Google will be communicating with your device constantly. That sounds bad on the surface, but the data they receive is not extremely threatening. Their services are severely restricted as an application, and they cannot see everything else your device is doing. They will know the IP address to your device at all times. Does that matter to you? When you are on a cellular network, you are probably sharing the same public-facing IP address with many other users on the same network. When on Wi-Fi at your home, you would only be sharing a VPN-protected IP address (if you followed my guide on a home firewall). I don't see either as a huge risk, but you might. I discuss this further when explaining proper VPN usage.

Google may receive some information about the applications you are using, but they will not receive any content from the notifications. They cannot read your email. The transmissions are encrypted. Since you are not required to create a Google account for this usage, there is no easy way for Google to attach your activity to a specific account. They will maintain connection logs, but not associated to a specific Google account. The ability to do all of this without an account reduces the overall privacy risk.

What do I do? I do not enable push services. I simply do not need them.

What do my clients do? Practically every client with a GrapheneOS device has push services enabled. This allows them to stick with the device and carry on with their lives. Their privacy and security are way beyond what they would receive with a traditional Android or iOS device, even with push services enabled.

Please remember there is no elitism here. It is more important to successfully take small steps toward a moving privacy goal than to fail from trying to do everything perfect from day one. For most readers, I believe enabling push services is appropriate. You can always disable them later. However, I believe they should be enabled before installing applications if you plan to go that route. Activating push services after weeks of usage without them will work, but I have experienced hiccups with some app notifications.

Now you must make a choice. If you want to enable push services, continue through this chapter. If you do not, skip to the next.

Enabling these features is quite easy thanks to the GrapheneOS application. Swipe up to see your application drawer and tap "Apps". These are the applications included from GrapheneOS. Click the "Google Play" services option and install it. All three options must be activated, but installing "Google Play Store" should enable all. When prompted, allow installation with default network permission for all three options.

When complete, click "Settings" below "Google Play services". If that is not visible, navigate to "Settings" > "Apps" > "See all..." > "Google Play services". Tap "App battery usage" and change it to "Unrestricted". You should now see a new notification at the top of your home screen letting you know that "Sandboxed Google Play is running". I don't need to be reminded of this, so I navigate to "Settings" > "Apps" > "See all ..." and tap the three dots in the upper-right to "Show system". I then tap "GmsCompat" > "Notifications" and disable all.

That's it. You now are ready to install applications and receive the benefits of push services without allowing Google unfettered access to your entire device. If you change your mind, you can disable all three options by opening each; clicking the three dots in the upper-right; and selecting "Uninstall". Your device will be Google-free again.

Once you begin installing applications, as explained in the next chapter, you will probably want to modify various notifications options. Anytime you receive an unwanted audible, visual, or vibrating notification, control this by going to "Settings" > "Apps" > and selecting the desired application. You can then open the "Notifications" option to adjust every setting as desired. This can take some time to get perfect, but the final result is worth the effort.

# CHAPTER SIX

## APPLICATION INSTALLATION

A default GrapheneOS installation does not include any Google services or the Google Play Store which is used to install third-party applications. If you enabled Google's push services, you technically possess the Google Play Store application, but it will not install anything without an associated Google account. We should not compromise our privacy by relying on Google for our apps. Instead, we will use better options. I always start with the installation of F-Droid.

F-Droid is an app store and software repository for Android. It presents a similar function to the Google Play Store. The main repository only contains free and open source apps. Applications can be browsed, downloaded and installed from the F-Droid app without the need to register for any account. The following installs the main F-Droid app onto your GrapheneOS device.

- Launch the Vanadium browser.
- Navigate to [f-droid.org](https://f-droid.org) and click the "Download F-Droid" button.
- Confirm the download and click "Open" at the top of the screen.
- If prompted, click "Settings" and enable "Allow from this source".
- Confirm the installation of F-Droid.
- Open the F-Droid application and confirm any warnings.
- Click "Don't Allow" for notifications.
- Swipe down from the top and fetch any F-Droid updates available.
- Tap "Updates" to install any pending updates.
- If prompted, repeat enabling of "Allow from this source" settings.
- Reopen the F-Droid application.

You now have a substitute app store which is not powered by Google. Many of the open-source applications we will use will come from this repository. This device is more private and secure than any stock unit which could be purchased from a retailer. Unlike a traditional iOS or Android phone, a user account is not required in order to download apps. If ever prompted to add a Google account, avoid or "skip" the option. This way, there is no single Google or Apple account which can be tracked, archived, and abused. Again, by default, GrapheneOS transmits no data to Google. Eliminating these privacy threats provides great benefits.

The installation effort can seem overwhelming, but is usually only a one-time event. Updates are automatic by default and pushed to your device often. You may notice them within the notification menu, and you may be prompted to reboot to finish installation. Along with F-Droid, I recommend the application Aurora Store. This is an unofficial client to Google's Play Store. You can search, install, and update apps. You can also spoof your device, language, and region to gain access to the apps which are restricted in your country. Aurora Store does not require Google's framework. With Aurora Store, you can install all of the mobile apps mentioned throughout this book. Aurora Store can be installed through F-Droid by conducting the following.

- Tap the "Latest" icon within F-Droid and tap the search icon.
- Search "Aurora Store" and tap "Install".
- Allow the installation to complete and open Aurora Store.
- When prompted, accept their Terms of Service.
- Tap "Next" four times to navigate through the screens.
- Tap "Grant" on the first option, then "Allow".
- Tap "Grant" on the second option, tap "Aurora Store", and "enable access".
- Tap the back arrow twice to return.
- Tap "Grant" on the third option, then "Allow from this source".
- Tap the back arrow and then "Finish".
- Tap "Anonymous" mode, which prevents Google account requirements.

The previous instructions were written for the initial release of this guide in February 2023. In May of 2023, the Aurora Store began having problems. Several people reported the inability to install new apps or update existing software. Some people reported that logging out and back in corrected the problems for a few days. As I write this on May 30, 2023, the developers behind Aurora Store acknowledge that the anonymous Google accounts embedded into the app which allow it to fetch details from the Google Play Store were actively being blocked, causing the app to malfunction. As of today, everything seems to be working, but for how long?

If you followed the above installation and are able to download and install applications through Aurora Store, then you are all set and can **proceed to the next section** titled "Controversy". If you are having issues, you are not alone. The following text will walk you through every option you have. Even if Aurora Store stays working fully again, I plan to leave this updated instruction within this guide for future problems. First, you should understand all application installation and upgrade options when Aurora Store is malfunctioning. The following are presented in order of least invasive to most.

- **Wait:** Many times, the issues with Aurora Store will correct themselves after any embedded Google accounts have had time to reset themselves. The issues we have seen in May 2023 have presented more downtime than in the past, but the Aurora team is always monitoring for issues.
- **Workaround:** If you installed Aurora Store before any issues began, you can likely still use the program to install or update software, but the method will be slightly different. I explain this in a moment.
- **Manual:** You can always download any desired applications from a source such as [apkmirror.com](https://apkmirror.com); install the downloaded APK files; then repeat the process any time you think there might be an app update. This can be time consuming and annoying, and I do not do it, but no account is required.
- **Google Account:** If you are installing Aurora Store for the first time since these issues began, or reinstalled the app trying to correct things, you might be required to log in to a "burner" Google account within Aurora Store. You can log out of the account right away and continue to use the "Workaround" technique, but Google will receive very limited information about this one-time connection. I explain more in a moment.

Let's tackle the "**Workaround**" option first. This should only apply to people who installed Aurora Store BEFORE any issues began, but there is no harm attempting this with a new installation. We must add the option to open Google's Play Store links within Aurora store with the following steps on your device.

- Open "Settings" and select "Apps" > "Default apps" > "Opening links".
- Enable "Instant apps" and tap "Aurora Store".
- Enable "Open supported links" and click "Add links".
- Enable all options and repeat the previous link opening process.

If that last step does not allow you to add links, you may need to disable them from Google Play first. Conduct the following.

- Open "Settings" and select "Apps" > "Default apps" > "Opening links".
- Enable "Instant apps" and tap "Google Play Store".
- Disable "Open supported links".
- Open "Settings" and select "Apps" > "Default apps" > "Opening links".
- Enable "Instant apps" and tap "Aurora Store".
- Enable "Open supported links" and click "Add links".
- Enable all options and repeat the previous link opening process.

Anytime you click a link which would otherwise send you to the Google Play Store, it should now open that same link within Aurora Store. Let's test it. From your Vanadium browser application, conduct a search for "Signal Play Store", without quotes. The first result should be a link to the Google Play Store page for the secure messaging application Signal. Mine was the following.

<https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms>

Click the link to open the page. It should open the Signal application page within Aurora Store. You might see the full information about the application and a button which displays "Install" or "Open". If you do, you could install or open this application from this page. The page might also be blank. Either way, click the back arrow in the upper left. This should take you to the home page of the Aurora Store. Click the "Updates" button in the lower-right. If it presents outdated programs, you should be able to update them as normal.

Some readers might be able to use this workaround technique to install or update apps at any time. When installing new software, you would need to search the program and identify the Google Play Store page, and then click that link to open the installation option within Aurora. You can also search any application within vanadium; click the link; click the back button within Aurora; and update from there. This scenario is not ideal, and requires a few extra clicks, but may be enough to get you through whenever Aurora Store is having issues. If this does not work for you, I have other options.

If you only see blank pages during these attempts and no option to install or update any applications, then your instance of Aurora Store is broken. You likely no longer have any type of login token from previous usage, and Google is blocking access to its data without this authorization. The "**Google Account**" process is the most reliable option for you, but it is a bit invasive. This requires you to open the Aurora Store application and choose the "Google" account option. You would then log in to any existing Google account in order to permanently access the ability to install and update programs from within Aurora Store. This sounds like a huge privacy violation, and it may be inappropriate for you. Let's consider the harm it may cause.

If you have a Google account which is not associated to your true identity, it may be acceptable for this situation. If you log in to this account from within Aurora, Google will see that you are accessing the account from a Pixel device; collect your current IP address; and possibly generate a unique identifier associated with this connection and any apps downloaded. They will not see your cellular number, device serial number, MAC address, or anything else outside of the Aurora Store application. This data is limited, but there are risks.

If this sounds creepy to you and you have worked hard to create a completely Google-free device, this may not be appropriate for you. You should either wait for a full working solution from Aurora or manually install applications as explained next. If the Google account you have has absolutely no history or connection to you, I think the risk is very minimal. You do not need to leave the account logged in all the time. The biggest benefit of this method is that you no longer need to use Aurora's anonymous login addresses, which are often rate limited.

If you want to proceed, you would open Aurora Store; choose "Google"; and log in with the credentials from your burner Google account. You can then either leave the account logged in all the time and have a fully-functioning Aurora Store application, or immediately log out from menu > "Accounts" and use the previous "Workaround" tutorial any time you want to make changes. I confess that I do not use this technique.

Finally, you might consider the "**Manual**" method until issues with the Aurora Store are resolved. I prefer APK Mirror ([apkmirror.com](http://apkmirror.com)) for this, but there are other options, such as APK Pure ([apkpure.com](http://apkpure.com)). Within APK Mirror, search the application which you want to install or update. As an example, I searched Proton Mail and was presented numerous versions of the application. I always choose the top version, as long as it is not a "beta" or "alpha" test build. In this example, it was version 3.0.15.

This opened a new page and presented a "Universal" variant with a downward-facing arrow. I clicked the arrow which presented yet another new page for that program. I then clicked the "Download APK" button, which downloaded the file to my mobile device. When the download finished, I opened the downloaded APK file and allowed the installation to complete. If I had already possessed this application, the process would have updated my version to the most recent. If the versions were the same, the installation process would have made no change.

If you only possess a small number of mobile applications, then this manual method may be the best option for you. You will need to keep up with application updates, but you will no longer rely on third parties to do this for you. If you want the easiest long-term solution, then the Google account option may be best for you. There are some minimal privacy invasions, but none more than what Google would collect from usage of their push services, as previously explained. However, I would never use a Google account associated with my true identity for this purpose. Finally, some readers may choose to just wait things out. I warn you not to wait too long though. The risks associated with outdated software are greater than the privacy risks of seeking a solution to update your applications.

What do I do? I had already installed Aurora Store prior to May of 2023, so I just used the "Workaround" method until things were back to normal. Once a week, I searched for an app, such as Signal, and opened the link for the Google Play page. That forwarded me to that app on Aurora Store and allowed me to use the back button to navigate to the home page. From there, I clicked the "Updates" tab and update all apps. I could have also used this to install a new app if needed. This is the path I recommend for anyone who qualifies to use it when things do not work. Fortunately, as I write this, Aurora Store was functioning again.

If a client was setting up a new device today; they are using Google's push services; they have access to a burner Google account with no association or history to their true identify; and the anonymous login option within Aurora Store was not working, then I would have little objection to them going the "Google Account" route. If they wanted a truly Google-free device, then I would NOT recommend this.

I always attempt any app installations through F-Droid before Aurora. If an app is missing from F-Droid, I rely on Aurora Store. You can use the "Updates" menu of each app to make sure all of your installed applications stay updated. Make sure to keep Aurora updated through F-Droid in order to maintain functionality, especially when things are not working properly. I launch both F-Droid and Aurora weekly to fetch any pending application updates. I do not rely on the notifications of either app to prompt me for action.

## **Controversy**

Some advanced readers may be upset at my recommendation of F-Droid and Aurora Store for application installation. There are elite online communities which debate the security and trustworthiness of these two options, but none of them can provide a better solution. At least once weekly, someone emails me who is upset that I recommend F-Droid and Aurora Store when there is research proving they are "bad options". These messages almost always point to a single blog post in which the author claims these applications cannot be trusted. The complaints range from slow updates to a "Confusing UX", and seem motivated by ongoing disputes between members of both sides. The solution recommended by those against F-Droid is to install the official Google Play Store, log into a Google account from your mobile device, and download all applications directly from Google. I think that is an awful idea.



There is also a small portion of the community which believes we should avoid any software store and install all applications from the open-source APK files released directly by the services which make them. This sounds great in theory, but many of the apps we rely on do not offer this. Furthermore, keeping these apps manually updated would take hours out of every week. Therefore, I rely on F-Droid and Aurora Store to do this for me and my clients in the best way currently available.

I am not naive and I trust nothing completely, including app stores or apps themselves. Malicious apps make their way into every repository, even the Google Play Store. This is why I only install the trusted and vetted applications which I truly need and never experiment with new services from my clean device. When we adopt a mobile device and rely on it for all communications, we never know every detail happening behind the scenes. We have to do the best we can while trying to live life outside of these debates. In the past five years of relying on these alternative application installation stores, I have never had a security scare. However, there are risks in everything we do in life, including our choices within mobile devices. Make the best decisions you can.

Let's pause and digest what we have accomplished. Our phone possesses the basic communications technology we need for daily use. It does not share any data to Google or Apple. An account is not required to download applications; therefore, an account does not exist to collect and analyze data about our usage. There are no embedded cloud storage options which can accidentally be enabled. This is a huge feature for most clients. This minimal device encourages us to return to the original intention of a mobile phone: communications. Finally, we can begin installing our favorite applications.

Selecting applications is a personal choice, so I will simply identify the software I use on my device, and those for most of my clients. I will explain some of these applications, including customizations, within the following chapters.

Proton Mail	Wire	Standard Notes
Proton Calendar	Element	AntennaPod
Tutanota	MySudo	Simple Notes
Signal	Sipnetic	Simple Voice Recorder
Molly	KeePassDX	

When you install any application, notice that GrapheneOS often prompts you to provide a network connection to that app. Most apps rely on internet connectivity, and the default access is appropriate. However, you might install some apps which never need network connectivity. This could include home screen launchers, local music players, voice recorders, etc. We will revisit this later and check our app settings, but block anything which you know never needs to access the internet.

Most applications should install without issues, but nothing is perfect. You may search for an app within Aurora and be unable to find it. In 2022, both MySudo and Privacy.com were not indexed within the native search feature (but did appear while writing this chapter). However, that does not mean we cannot install "hidden" applications from within Aurora. They are actually present if we know the exact URL,



but that is unlikely. There are two options for installing applications which are missing from Aurora's search.

The first is to visit the company's website, such as Privacy.com, from the mobile device and tap the button to install the app via Google Play. This should navigate you to the installation option for this app within Aurora. If that does not work, repeat the following changes which we made during the previous tutorials.

- Open "Settings" and select "Apps" > "Default apps" > "Opening links".
- Enable "Instant apps" and tap "Aurora Store".
- Enable "Open supported links" and click "Add links".
- Enable all options and repeat the previous link opening process.

If that last step does not allow you to add links, you may need to disable them from Google Play first. Conduct the following.

- Open "Settings" and select "Apps" > "Default apps" > "Opening links".
- Enable "Instant apps" and tap "Google Play Store".
- Disable "Open supported links".
- Open "Settings" and select "Apps" > "Default apps" > "Opening links".
- Enable "Instant apps" and tap "Aurora Store".
- Enable "Open supported links" and click "Add links".
- Enable all options and repeat the previous link opening process.

You should now be able to click a "Download from Google" website link and be forwarded to the appropriate page within Aurora Store instead. If you encounter a desired application which does not possess a link on their home page, search through the Google Play website. When you find the desired link, tap and open through Aurora.


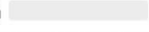
If this all fails, go to "Settings" > "Networking" within Aurora and enable "Insecure Anonymous Session". Log out of Aurora, close the app, open it, and log back in. If desperate, download the desired application's APK file from [apkmirror.com](http://apkmirror.com) or [apkpure.com](http://apkpure.com) and install it manually. This should populate the app within Aurora for all future updates. Note that some apps are restricted to specific countries which may prevent access.

Once you have installed all of your desired apps, navigate to "Settings" > "Privacy" > "Permission Manager" and consider these options. By default, some apps may already have permission to access your camera, microphone, or other hardware features. Communication apps obviously need access to your microphone, but a calendar does not. Consider modifying everything in this menu to your specifications. As an example, I disabled all "Body Sensors" access and severely limited my location, microphone, and camera access. I also disabled all "Nearby Devices" associations, which allows the use of wearable devices, such as a smart watch.

Next, we should take a look at our NextDNS portal to see where we may have data leaking from our applications. Log in at [nextdns.io](https://nextdns.io) and click the "Logs" tab. This should present any connection being made from your device over the past hour. This will vary for all readers, as we all have unique applications and usage. Mine appeared as the following.

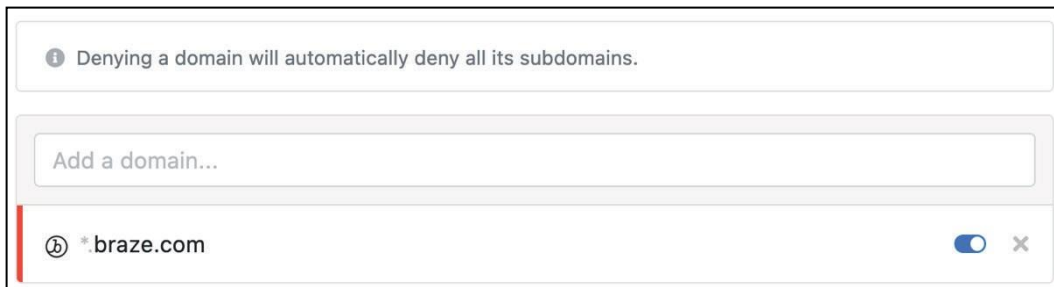
 api.protonmail.ch	  14 minutes ago
 mail.tutanota.com	  25 minutes ago
 chat.signal.org	  Tuesday, February 7, 2023 10:46 AM

This seemed appropriate. These are connections from Proton Mail, Tutanota, and Signal, likely checking for any incoming communications. So far so good. Next, I launched MySudo, an app which is explained later. I immediately saw the following. within NextDNS.

 sdk.iad-03.braze.com	  a few seconds ago
 eventgw.twilio.com	  a few seconds ago
 ers.twilio.com	  a few seconds ago
 messaging.service.anonymome.com	  a few seconds ago
 api.service.anonymome.com	  a few seconds ago

Let's work our way from the bottom up. The two anonymome connections are to the parent company of MySudo. They are checking for incoming messages and any other notifications. The two Twilio connections are directly to the provider of MySudo's VoIP numbers, so those are also fine. The top connection bothers me. Although I had previously disabled "Allow Anonymized Crash Reports" and "Allow Anonymized App Analytics" within MySudo at "Settings" > "Privacy", the service is still making a call to braze.com. Braze is an online analytics company which services use to identify issues with their apps and other products. An executive at MySudo confirmed that this connection is not sending any user information to Braze, and is solely used to push any notifications, such as a product update, to the user. However, I don't like to take any chances.

I don't want any app I use to ever send data to Braze, so I will block it within NextDNS. I clicked "Denylist" within the NextDNS portal and added the domain of braze.com. My entry appeared as follows.



I then reloaded the Logs tab and saw the following.



From now on, as long as I am connected to my custom NextDNS address within my device, no data will ever be sent to the Braze network. Let's work through another example. I rely heavily on masked virtual credit cards from Privacy.com. I opened the app and then refreshed my NextDNS Logs list. It displayed the following.



The connection to privacy.com made sense, and I know they use Azure servers to distribute data, but I do not like that connection to bugsnag.com. Bugsnag provides error reporting to the creators of mobile apps. A small file is sent from your device to Bugsnag, and then forwarded to the app maker. It often includes specific information about your device which I do not want to share. Fortunately, my default block list within NextDNS had already blocked it. If it had not, I would have added that domain to my deny list.

With my minimal GrapheneOS configuration, I could not find anything else which needed blocked. I encourage you to open NextDNS, refresh the Logs, and open one app at a time on your device. Research any connections made and identify those which you may wish to block. Test new applications the same way.

## Mapping Applications

There are no map applications included with GrapheneOS. You could install Google Maps from Aurora Store and possess the standard functions. However, you are now sharing data with Google again. I currently recommend either Organic Maps, Magic Earth, or OSMAND+. None of these will provide the level of location detail or traffic conditions as Google Maps, but these do not share constant data about your activities with Google. Let's compare each.

Organic Maps and OSMAND+ are completely open-source applications and are free to use and modify. Magic Earth is free, but not open source. All allow you to download maps for offline usage, but Magic Earth makes this easier by state. Organic maps and OSMAND+ do not include reliable updated live traffic information, but Magic Earth does. However, it is not as reliable as Google Maps.

Lately, I prefer Magic Earth. I download all street maps of the United States to my device (>16GB). When I need to find a location or navigate to a specific address, no data is shared about my trip, and I can disable connectivity if the route is extremely sensitive. The application and maps have helped me tremendously when cellular service was unavailable in remote areas. If you truly need Google's navigation, then you could install it within a secondary profile, as explained later.

## Calendars

Much like everything else we have become accustomed to having on our old devices, there is no native calendar synchronization services embedded into GrapheneOS. I currently use the Proton Calendar app for all of personal and work schedules, and access is included with my email plan. Tutanota users also have a secure Calendar solution within their native app and web service.

## App Security

In episode 291 of my podcast (Mobile App Privacy & Security), I explained how a client was the victim of an iPhone theft which resulted in access to her bank accounts and email messages. Most of that scenario does not apply to our situation, but there are some vital lessons to be learned. Imagine someone stole your new device. Now imagine they were previously watching you at the bar while you were entering your PIN to unlock the device, and memorized your code. This is more common than one might think. What could they get? Most often, quite a lot.

Once your device is unlocked, most apps will open without any further authorization. Your secure email, 2FA tokens, text messages, and media content are immediately available to the thief. Worse, any financial apps will probably load directly to your account to allow transfers. Because of this, you might consider additional security settings within most applications, as explained within the following.

- Proton Mail allows you to set a custom PIN which is required to open the application. If I steal your device after watching you enter your PIN, and I then open Proton Mail, I must know the unique PIN you set for that app. This prevents me from accessing your email to conduct password resets across all services.
- Standard Notes allows a unique PIN to be set within each note. Since I use this product for my 2FA tokens, a thief would need to know my PIN for that note, which is not the same as my PIN to unlock the device.
- MySudo also offers a unique PIN to unlock the app, which could prevent SMS-based 2FA from being abused.
- Privacy.com allows a unique PIN to open the app, which would prevent current and new virtual credit cards from being abused.
- Unfortunately, Signal and Tutanota only allow you to use your device PIN to open the apps, which could be problematic. Wire allows a custom PIN.

I confess that my custom PIN to open each application on my device is the same, but it is not the PIN to unlock the device itself. This way, I must only remember two PINs. Regardless of your per-app settings, we can make things more difficult for a thief. Always cover your screen when entering your PIN in public, and a complex passcode could make it more difficult to remember the PIN as seen from afar. My PIN is 20 characters.

If my device was stolen while overseas, I could access my data-only eSIM configuration from another computer to disable network access, as explained later. I could also connect to my NextDNS account and block all activity, as previously explained. This would prevent Wi-Fi or cellular connections from being used to conduct any nefarious activity from my device or applications (unless the thief knows how to change the DNS settings). Think about your own risks and determine if you need to make additional modifications to harden your own device.

There will be much more detailed discussion about many applications throughout the rest of this book. For now, I will assume you have installed and configured most of your desired applications. It is now time to activate private cellular service.

# CHAPTER SEVEN

## CELLULAR SERVICE

Warning: This chapter will be heavily focused on readers who live in the United States. However, the overall lessons can be applied globally. The following two chapters include numerous international considerations.

Now that you have a secure and private mobile device including proper configuration and desired applications, you need cellular service. While I do have clients who forgo a cellular plan and rely solely on publicly-available Wi-Fi, they are a rarity. Most of us want a connection available at all times for our communications.

As stated previously, every cell phone is a tracking device. There is no way around that. Therefore, I insist that the cellular service for myself and my clients is established in an alias name. When the various cell towers track and document the location and communications of "John Doe" at all times, I care less about the invasion.

Most people in the U.S. who have a cell phone also possess a contracted plan. They walked into a wireless provider's store and purchased their device and plan together. They were provided a steep discount on their favorite device in exchange for a two-year contract. That contract required a soft credit pull and a copy of their license. Their overpriced plan made the actual cost of their device \$1,000 instead of the retail rate of \$500. Their name, address, number, and DOB will be public data after the next breach, and their name is now publicly associated with that number forever.

When you place a contracted mobile plan in your true name, the data is almost immediately shared with third parties. Caller ID databases connect the data and allow anyone with \$0.03 to query the information. I explain several paid and free options for finding the name associated with almost any cellular telephone number in my *OSINT Techniques* book. There is absolutely no privacy in these situations.

In order to obtain a cellular plan in an alias name, you will need a prepaid provider. There are several options, and I present those which I use consistently. Typically, I avoid plans with the carriers directly. Instead, I rely on resellers, as explained next.

### **Mint Mobile**

In major U.S. metropolitan areas, I use Mint Mobile as the provider. Mint is a T-Mobile reseller, and only offers prepaid plans. I choose them because they are very affordable, do not require user verification, and allow prepayment up to a year. At the time of this writing, the lowest monthly unlimited plan was \$15 including a free physical SIM card or programmable eSIM. I only need the data, as most clients will never use their real T-Mobile issued number for calls or texts. This plan includes 4 GB of monthly high-speed data and unlimited data throttled at a lower speed after the 4 GB.

You can obtain physical SIM cards from Mint directly from their website, Amazon, or BestBuy. The cards are free if you purchase a package directly from Mint and \$1 to \$5 for two cards if you purchase from Amazon. I purchased dozens of 2-packs from Amazon using an anonymous account and shipped to an Amazon Locker, but this may be overkill for your needs. If you only need one or two devices activated, and prefer a physical SIM card, I recommend either purchasing the Mint Mobile Starter Pack from a BestBuy location or have Mint mail you their cards at no cost. The following are my recommended strategies, in order of privacy.

- BestBuy: If you are near a BestBuy store, this is the easiest and most private option. Most stores carry the "Mint Mobile \$5 Prepaid SIM Card Kit" with a SKU of 6310600. At the time of this writing, the cost was \$1.00 and each included \$5.00 in credit. I have been able to purchase dozens at a time.
- Mint Mobile: Mint will happily send a SIM card to any U.S. address and in any desired name for free. This obviously creates a digital trail to a physical location, but I have used P.O. Boxes, CMRA addresses, and even General delivery to receive these cards in an alias name.
- Amazon: Purchase an Amazon gift card with cash from a physical store, such as a grocery store. Create a new account on Amazon using alias information and an address of a hotel near your location. Apply the gift card to the account and purchase the Mint Mobile Starter Pack. Choose a nearby Amazon Locker for the delivery address. Once your cards arrive, obtain them from the locker. This will always be more difficult than the previous option due to Amazon's fraud detection systems which may block your order.

None of these apply to most of my clients any more, because I rarely purchase any physical SIM cards for them. Instead, I rely on their device's eSIM option. Before we proceed, we should understand the benefits and inconveniences of each option.

Physical SIM cards are the traditional small chips which we slide into our mobile devices. Our Pixel options for GrapheneOS all include this tray, while newer iPhones do not possess them. The main benefit of the physical SIM is the ability to transfer it at any time. If you buy a new device or your current phone breaks, you can easily swap the card into another unit without assistance from the cellular provider. This is very important for those readers who switch phones often or have multiple devices used throughout the year.

Programmable eSIM cards are a newer technology which is also available in our Pixel GrapheneOS devices. No physical card is needed. Instead, the cellular companies provide either a text code to input or a QR code to scan. This programs all necessary data into your device and the eSIM within your device's hardware functions identically to a physical card. This is sometimes my preference for the following reasons.

- No shipment is required. I do not need to convince Amazon, Mint Mobile, or another online retailer that I am worthy of their card. I do not need to provide a shipping address for the package. My purchase will not be scrutinized and I do not risk the association of a physical address to my account.



- Multiple eSIMs can be stored. I can program multiple providers into my device and switch at any time. I can choose various providers based on my location. I can also reserve a provider for Wi-Fi access to the number without connecting to a cellular tower. I explain more on this later.
- eSIMs can be enabled and disabled without completely removing their function. If I want to use multiple accounts which only provide access through physical SIM cards, I must continuously remove and insert new cards. I must also carry multiple tiny cards with me at all times. I cannot tell you how many SIM cards I have lost over the past decade.

The inconvenience of eSIM programming is the inability to easily move the account programming to another device. Most providers allow this, but it often requires you to contact customer support to make it happen. Some companies limit this activity to once or twice per year. If you change phones often, this may not be a great option. If you plan to rely on your new Pixel device, then it should not be much of an issue. Programming eSIM connections also requires you to enable "privileged eSIM management" in the "Network and internet" menu, which requires the sandboxed Google Play Services to also be installed. However, you can disable and remove these options once the eSIM is programmed, if desired. Please note that reinstalling your operating system overwrites all of your settings. Only apply these once you know your device is how you like it.

### **Mint Mobile Physical SIM**

Let's start with the physical SIM option, as it is the easiest to activate without providing any real name or address for yourself. It is also the most globally-recognized format and requires no programming directly into the device. After you have obtained a Mint Mobile SIM pack, insert the card in to the device. Install the Mint Mobile app from Aurora Store on the device which you recently configured. This should be done away from your home. If possible, use public Wi-Fi.

After launching the app, choose the "Activate your SIM card" button and follow the directions. It will require the number printed on the card. You will need to provide a name and email address for the registration and physical address for the billing. They will require a credit card for payment toward a new account. Let's discuss each.

Mint Mobile does not validate any information, so a random alias name is fine. I have never seen them block privacy-respecting email addresses from Proton Mail or Tutanota, so you should be fine there. I have seen them scrutinize new accounts when registering behind a VPN, which is why I recommend public Wi-Fi. I have witnessed the Mint app require you to enable location services in order to determine if you have cellular coverage in your area. I do not object to this since I would never activate from my home and they will know my location based on cell towers anyway.

The billing address is important. They do not scrutinize any information provided, but we want to make sure we are complying with the law. Most cities, counties, and states within the U.S. apply various taxes toward cellular services. These taxes were originally



intended to pay for emergency services related to 911 calls, but now they seem to be used for anything. I encourage you to identify a hotel within the area of your primary usage and provide that address. This way, you are paying the appropriate taxes on your account. Since Mint will have access to the locations where your device uses their services, they would know the general area where you are anyway.

Mint allows the use of masked cards, such as virtual card numbers from Privacy.com. These can be closed at any time if you have an issue stopping the billing. If you do not have a masking service, secondary credit cards, as explained in *Extreme Privacy, 4th Edition*, have worked well. It is impossible to be completely anonymous here, so my focus is on as many privacy layers as possible.

After you select your desired plan and make successful payment, you are done. You will be issued a cellular number within the area of your provided address and you will have immediate access. An account will be generated and you can use the mobile app to monitor your usage, renew plans, etc. All of your usage is documented forever, but they do not know your name. When data is leaked, it will have no direct impact on your name, your home address, or your communications.

### **Mint Mobile eSIM**

If you do not possess a physical SIM or do not want your account associated with a true physical shipping address, you can register the account to your eSIM slot of your device. Upon launching the Mint Mobile app, work through any introduction screens after selecting the "Try" feature. You will need to provide the same details mentioned in the previous section, so be prepared for that. At the time of this writing, I was offered a free 7-day unlimited trial if I registered via eSIM.

Once complete, Mint will generate a new eSIM option which needs to be registered to your device. I always choose the "Enter QR code manually" option when prompted. This will present a long string of characters which begins with "LPA:". Copy this text to your clipboard via the "Copy" button and navigate to your device's "Settings" > "Network & internet". Enable "Enable privileged eSIM management". Note that you must have the previously-explained Google push services activated for this task, but you can remove them after the activation is complete. This applies to any device which is registering or switching an eSIM.

Tap "SIMs", "Download a SIM instead", "Next", then "Yes". When prompted to scan a QR code, click "Need help" then "Enter it manually". Paste the code previously copied and click "Continue". Click "Download" when prompted and "Settings" when finished. Enable the "Use SIM" toggle and confirm the choice. Enable "Mobile data" and "Roaming". You should now have cellular service, and you never provided a true name, DOB, or physical address. The Mint Mobile app will display your trial cellular number, which is also included in the welcome email.

However, we are not done. This is where things currently get murky with Mint. Creating this free trial is easy. Your device will have access for a full 7 days. Renewing

is surprisingly difficult. At the end of my trial, I opened the Mint app on my device and I was offered another free 7-day trial. Since I had no Mint account which I could log in to, it seemed to not know that I was already a member. I had to contact support via their online chat and have them send me my "activation code". Only then could I access the Mint app, renew my service, and create an account. This may be fixed by the time you read this.

If coverage is acceptable, you can purchase an annual plan for \$15 monthly. Is the eSIM process worth the headache? Only you can decide that. I confess I rely on a physical SIM since I switch devices and test new features often. My clients typically receive an eSIM which requires no shipment or true address. Either path provides the exact same service.

Overall, the new account creation process and service registration with Mint Mobile is less scrutinous than other resellers. They care more about being paid than verified identity. This is why I prefer them. However, there is a concern. In January 2023, rumors appeared about Mint Mobile being acquired by T-Mobile. We do not know if the sale will happen or if the two companies will merge. Even in a worst-case scenario, I believe Mint Mobile would stay around and offer the same features as described here. Since Mint is using T-Mobile's network anyway, I don't see much more invasion if T-Mobile becomes the parent company. You may feel differently, so I will present a few additional options for your consideration.

T-Mobile currently offers three months of free service to test their network directly, however this probably will not work for your needs. When installing their app to an unlocked Pixel without any carrier affiliation, you are told that you do not qualify. They are focusing this offer toward people already locked into another carrier.

## **Tello eSIM**

Tello is also a T-Mobile reseller with more customizable plans. On their web page at [tello.com](https://tello.com), you can choose the amount of monthly calls, text, and data required, then pay a price appropriate for your needs. If you never plan to use the cellular number provided by the carrier, a 5 GB monthly data plan is \$15. If you only need 2 GB for basic data communications, it is only \$10 monthly. If you only need calls and texts with no data, you can pay \$5 monthly.

If you purchase a plan with Tello, do it from a desktop computer. Tello will issue you a unique QR code which can be scanned from your device for easy eSIM programming. At the time of this writing, I took the following steps from the device, after purchasing a monthly account from the website, providing an alias name, address, and masked payment.

- Navigate to "Settings" > "Network & internet".
- Ensure "Enable privileged eSIM management" is enabled.
- Tap the "+" next to "SIMs".
- Tap "Download a SIM instead".

- Click "Next".
- Use the device camera to scan the QR code from the Tello site.

Tello then finished the eSIM installation and I possessed service. The more I use Tello for clients, the more I like their options. Being able to pay for only one month at the reduced rate is great, and having options to save money is a huge benefit. I rarely go over 1 GB of data every month since I only use my device for communications, and never entertainment. I could probably get by with their \$6 plan.

### **Secondary Account**

You may now be thinking about the possibilities with two plans. You could possess service through a physical SIM card and a secondary account associated with the eSIM slot. Why would anyone do this? I have an example to share. I have a client who relies heavily on a banking app for mobile check deposits. She is self-employed and receives paper checks for payment to her CMRA address weekly. She does not have a local bank branch in her area, and must take photos of the checks within her bank's app in order to deposit them. That app insists that a true cellular telephone number be associated with the account, and it sends a text message for authorization every time the app is opened. It refuses to use internet-based numbers such as Google Voice and others. In other words, my client must have access to a true cellular number every time she opens the app.

She has a Mint Mobile SIM card in her device which provides a cellular number to her. However, connecting that number to her bank account seems reckless for her threat model. She does not want a cellular account which possesses location data about her device at all times to be associated with a bank account in her true name. Therefore, she possesses a Tello voice and text account within the eSIM slot of the device. Whenever she needs to receive a text message to her Tello number, she enables the eSIM within her GrapheneOS device. The Tello account connects to a cell tower and provides her service. She opens the bank app and a text message is sent to her Tello number, appearing within her messaging app on the device. She logs into her banking app and then disables the eSIM.

This may seem extreme and unnecessary. However, it works well for her. The Tello number connects to the provider once a week from a designated physical location, but is not otherwise tracking her every move. She never gets locked out of her bank account because she has attached a true cellular number to the account. This costs her an additional \$5 monthly, but is justified for her usage.

### **AT&T and Verizon Resellers**

T-Mobile is not the only game in town. If you prefer the networks of AT&T or Verizon, you can obtain service anonymously with them as well. Before you commit, consider the next section about Wi-Fi calling. If you prefer AT&T, I believe the best option is Red Pocket ([redpocket.com](http://redpocket.com)). You can select an annual plan with unlimited calling, text, and data for \$15 monthly. They provide an eSIM installation similar to

the Tello scenario previously explained. They typically offer a free trial, but registration was disabled at the time of this writing. If you prefer Verizon, I believe US Mobile ([usmobile.com](http://usmobile.com)) is the best option. An unlimited talk and text plan with 5 GB of data is also \$15 monthly. They also provide an easy eSIM installation which provides immediate service and a free 10-day trial to test their network. If Wi-Fi calling is important to you, as explained soon, then you need a T-Mobile reseller for your unlocked phone.

## **SIM and eSIM Disabling**

One of my favorite features of GrapheneOS is the ability to not only disable an eSIM, but the possibility of also disabling the physical SIM via software. Most Android devices tell you to remove your physical SIM card if you want it disabled. This is a burden due to its small size and the need for a tool to open the SIM tray. GrapheneOS provides an option to disable the physical SIM card via toggle within the SIM card's settings page (some older devices do not allow this, but the 6a does). This allows me to completely disable all physical SIM and eSIM accounts, regardless of airplane mode. If I should accidentally disable airplane mode while near a sensitive location, my SIM and eSIM connections are not enabled. This does not prevent my device from communicating with nearby towers, but it does prevent the connections from being associated with my cellular accounts. Right before I ever enter airplane mode, I quickly disable any active SIM or eSIM options.

## **Wi-Fi Calling**

Wi-Fi calling is a double-edged sword. This feature allows you to make and receive calls and texts through your cellular carrier number while in airplane mode and connected only to Wi-Fi. Why would you need this? A few moments ago, I shared a scenario where my client needed to receive a text message from her traditional cellular number. As long as a physical SIM or eSIM is active, this can be done over Wi-Fi without any connection to a cellular tower. Every provider has different rules for this, but I will walk you through a common scenario.

Within Mint Mobile, you must have Wi-Fi calling enabled. You can do this through the app (preferred) or through their website. This will require that you supply the address provided during registration as your emergency location. This is only used if you call 911. The operator will see the address associated with the device, but will also see the true approximate location of the phone. Once Mint has confirmed that your provided address truly exists, they will enable Wi-Fi calling on your account. You can now enable the feature within GrapheneOS with the following steps.

- Navigate to "Settings" > "Network & internet" > "Sims".
- Select your desired SIM if necessary and tap "Wi-Fi calling".
- Enable "Use Wi-Fi calling" and c
- Change the "Calling preference" to "Call over Wi-Fi".

You should now be able to make traditional calls from your carrier provided number

within the native GrapheneOS dialer app while on Wi-Fi. Text messages can be facilitated through the messaging app. However, should you do this? Calls and texts made this way are logged in your cellular account forever. I personally never need this, as I rely on Voice over Internet Protocol (VoIP) numbers, as explained next. However, it does have some advantages. If you do not have VoIP options, or cannot access them, calls unrelated to your identity might be fine. This could include calling businesses to find out their hours or locations. I have used this before to make a restaurant reservation in an alias name. I see little harm there.

This could also be used as part of the banking scenario previously explained. If you are forced to use your true cellular number in order to access your banking account, you could conduct the entire transaction while in airplane mode and only connected via Wi-Fi. There are many options here, and only you can decide the appropriate path. Will using your true cellular number only with your bank create a connection between the two? Of course. Will that connection become public? It is very unlikely. Even if it did, you could always establish new service. Overall, I want to make sure my clients can continue a normal life, even at the cost of minimal exposure.

Please note that Wi-Fi calling through AT&T and Verizon reseller plans will likely not work with your new device. They only allow this feature on devices branded for use with their specific networks. It is typically blocked on unlocked devices. Also note that Wi-Fi calling will drain your battery faster, as it is always listening for an incoming call. You should consider disabling it whenever not in use. I ask my clients to enable it when they need to receive a call or text to that number, and immediately disable it when complete.

If you never use your true cellular number, then you should never need to receive SMS text messages from your carrier. I have seen several online posts encouraging people to disable the messenger app completely in order to block spam messages or malicious incoming content. I do NOT recommend disabling all SMS text messages. If someone would ever attempt a SIM swap or other nefarious activity within your account, a text message from your carrier could alert you to the issue.

We have discussed a lot. Let's summarize a few things. What do I do? Currently, I rely on a Mint Mobile physical SIM card within my device. I switch devices often and I am constantly testing new things. Therefore, the physical SIM makes the most sense for me. I maintain a few data-only accounts within the eSIM profile settings, which are explained in Chapter Nine. I have several VoIP options, which are explained in the next chapter. I pay \$15 monthly through an annual subscription.

What do my clients do? Almost all of my clients possess a Mint Mobile eSIM as their sole provider for data, voice and text. Most of them do not ever use the associated telephone number and rely on the VoIP options explained within the next chapter. Some possess a secondary account for calls and texts via eSIM which they can enable whenever needed. A few enable Wi-Fi calling features for either of their official telephone numbers, but only for use as true cellular text messages required for a minimal number of financial accounts.

# CHAPTER EIGHT

## VoIP SERVICE

Now that you have a new device with a new cellular plan (or plans), you could start using these accounts for any traditional telephone calls without any further action. However, I never want to rely on the number associated with my mobile device for my daily communications. Therefore, we will need a way to make and receive standard telephone calls and text messages without using our cellular plan. Within GrapheneOS, I rely on an application called Sipnetic and various Voice over Internet Protocol (VoIP) providers for all telephone calls. Before we configure our devices, let's understand the reasons we should be careful about true cellular number usage.

- When you make calls and send text messages through your standard cellular number, there is a permanent log of this activity stored by the provider of your service. This log identifies all of your communications and can be accessed by employees, governments, and criminals. I have witnessed call and text logs be used as the primary evidence within both criminal and civil trials.
- Your cellular telephone number is often used as a primary identifier for your account. If I know your number, I can use this detail to obtain further information such as location history of the mobile device. Your cellular provider stores your location at all times based on the cell towers to which you connect. I can abuse court orders to obtain these details or hire a criminal to breach your account. In past years, we have learned about the ability of bounty hunters to locate mobile devices in real time by simply knowing the cellular number. No court order was required. Journalists have been able to track people's movements for years.
- Cellular telephone numbers are prone to SIM-swapping attacks. If I know your primary number, I can take over your account through various tactics and become the new owner of the number. I can portray you and receive communications meant for you. If you used that number for two-factor authentication, I now have the second factor.
- When you give your telephone number to your friends and family, they will likely store it in their contacts and associate your name with the entry. Someone will then download a nefarious app which requests access to the contact list, sending the contacts to online databases which can be queried. We have seen this with several apps in the past, including caller ID services such as TrueCaller and Mr. Number, which shared private contact details with the world. Have you ever received an email from LinkedIn asking you to connect with someone you knew? This happens when that person agrees to share their contacts, including email addresses and telephone numbers, with the service. Twitter also wants to obtain these details from any members willing to share them. It only takes one instance to make your cell number publicly attached to your true name.



Using VoIP numbers eliminates much of the concern of these threats. Consider the following.

- VoIP calls and messages are also logged within the VoIP provider's portal. However, we have more control of this information, and possess options to permanently purge content whenever desired.
- VoIP communications do not possess the same location details as cellular connections. While the VoIP provider might possess an IP address for the connection, there are no cellular towers which provide exact GPS coordinates. If you break into my VoIP account, you will never learn my true location.
- Illegally overtaking a cellular account is trivial today. It can be done within an hour. Porting a VoIP number into another provider can take over a week, and notification of this action will allow you to stop it. Whenever I am forced to use a telephone number for two-factor authentication, I always prefer a VoIP number over a cellular account.
- You cannot stop your friends and family from sharing your telephone number with abusive applications and services. If they only know your VoIP number, there is less risk. Once a VoIP number is publicly leaked with association to your real name, you can easily change it if desired. If you have multiple VoIP numbers, you can isolate them for various uses. When the world knows a VoIP number belongs to you, it cannot be abused in the same way cellular numbers can. Again, VoIP numbers cannot share your location.

The solution to all of this is to never use a true cellular number. Instead, we will only use VoIP numbers for all calls and standard text messages. In the following pages, I explain how to configure various VoIP services for telephone calls and SMS text. My goal is for you to create your own VoIP product which allows you to make and receive telephone calls on your new secure device at minimal cost. Furthermore, the numbers will be in your control. You will not need to maintain access to a Google account in order to enjoy the benefits of VoIP calls.

This section is technical, but anyone can replicate the steps. As with all online services, any of these steps can change without notice. It is probable that you will encounter slight variations compared to my tutorial during configuration. Focus on the overall methods instead of exact steps. The following explains every step I took in order to create my own VoIP solution with Twilio. Afterward, I present other options which may be more appropriate for some readers. Please read the entire chapter before making any decisions.

Before we dive into various cellular providers, we must take a quick detour and discuss domain registration. I encourage you to digest this next portion before moving on. The steps you take now might make everything much easier later.



## Domain Registration

In past writings, I explained ways to use anonymous email forwarding services and temporary access providers when registering for online services. I supplied tutorials for providing these masked and disposable addresses to various services to protect our privacy. Today, I believe you should establish a new domain for use with your new private and secure device. Many privacy-focused email services are actively blocked by online providers. If you try to use a SimpleLogin masked email address to open a new line of cellular service, it will probably be blocked. If you try to fool a VoIP provider into accepting a Mailinator or 10MinuteMail address for a new account, expect an immediate suspension. Because of this, I want to have unlimited acceptable email addresses associated with a recognizable domain as I continue to configure my device.

You could simply buy a new domain such as vandalay-industries.net and configure it for email access, but that may not be the best idea. Many cellular and VoIP service companies are now scrutinizing new accounts. If you register with a brand-new domain, they can see that. Many fraud prevention systems block any registrations from domains which were created in the past 30 to 60 days. Therefore, I prefer existing domains which have recently expired and been dropped from their registrar.

First, I navigate to [expireddomains.net](https://expireddomains.net) and then click the "Deleted Domains" tab. I then sort by the following categories until I see desired domain structures.

BL: Number of known backlinks

ABY: The first year the domain was seen at Archive.org

ACR: Number of Archive.org crawl results

Reg: Number of Top Level Domains (TLDs) which match the domain

Below is an example of a few random results. While two of these have some internet history, none of them look like a traditional business domain which would pass human scrutiny. I only acquire ".com" addresses for this purpose, as some providers block newer TLDs such as ".work".

Domain	BL	DP	ABY	ACR	Dmoz	C	N	O	D	Reg	RDT
esolo.top	0	0	2021	1	-	●	●	●	●	12	1.0 K
bhc331.top	0	0	-	0	-	●	●	●	●	0	0
bananad.top	2	0	-	0	-	●	●	●	●	4	226

Next, compare those results to the following, which I found by sorting by each category. Personally, I like "Rental-Bus.com" and "PrairieBoard.com". Either should pass as a legitimate company name. "PrairieBoard.com" could be presented as a board of directors' entity acting on behalf of practically any business. This may seem overkill for a mobile device, but I believe it is justified. After purchase, I reserve email addresses associated with this domain solely for use with my new device. When we get into VoIP providers, you will be glad you were proactive with this.

PrairieBoard.com	0	0	-	0	●	●	●	0	1	2 days	available
neamemories.com	0	0	-	0	●	●	●	0	0	Yesterday 19:44	available
FishyChat.com	0	0	-	0	●	●	●	0	0	Yesterday 19:45	available
BankerSkit.com	0	0	-	0	●	●	●	0	0	Today 19:04	available
Rental-Bus.com	0	0	-	0	●	●	●	2	2	3 days	available
biaidi.com	0	0	-	0	●	●	●	0	2	7 days	available
polisick.com	163	2	-	0	●	●	●	0	0	Yesterday 19:43	available
GoodStuffForGoodPeople.com	0	0	-	0	●	●	●	0	0	Yesterday 19:42	available

Next, I like to verify domain registration history through online services such as Whoisology.com. Many online services, especially VoIP providers, will replicate this type of search, so I want to know what they will see if their systems scrutinize a domain associated with a new account. Below is the entry for rental-bus.com. You can see that domain registration has been captured since April of 2013. If I were to purchase this dropped domain and use it with the email account I provide during purchase, I may appear much more legitimate than using a new domain which has never appeared online before.

Register Today

Whoisology is a searchable reverse whois / domain name ownership database with billions of records and tens of billions of data points.

### Historic Whois Lookups

September 2022*	June 2022
March 2022	December 2021
September 2021	June 2021
March 2021	December 2020
September 2020	June 2020
March 2020	December 2019
September 2019	
March 2019	June 2019
September 2018	
March 2018	December 2018
September 2017	June 2018
March 2017	December 2017
September 2016	June 2017
April 2016	December 2016
August 2015	June 2016
December 2014	December 2015
April 2014	April 2015
August 2013	August 2014
December 2012	December 2013
	April 2013

\* Indicates the archive you are currently viewing

Disabled archives

do not contain WHOIS data for this domain

## rental-bus.com

This is Whoisology's most current historical whois lookup for the domain name rental-bus.com. Click any of the records below (address, phone, email, etc) to perform a reverse lookup.

### Admin Contact

The Admin Contact is the person or organization who controls the domain.

Name	Masone, Michael (31) Changes: +0 ccTLD: 0
Org.	Global Charter Services (31) Changes: +0 ccTLD: 0
Email	itdept@busbank.com (26) Changes: +0 ccTLD: 0
Street	141 W JACKSON BLVD STE 300A STE 300A (26) Changes: +0 ccTLD: 0
Street 2	-
City	CHICAGO (782,250) Changes: +112,179 ccTLD: 15,095
Region	IL (279,576) Changes: +7,204 ccTLD: 22,978
Zip / Post	60604-2992 (37) Changes: +0 ccTLD: 1
Country	UNITED STATES (92,024,163) Changes: +11,201,481 ccTLD: 1,915,656
Phone	12035362106 (76)

### Other Details

These are technical details & related, connected to the domain.

Registrar Name	Network Solutions, LLC(5,500,237) Changes: -336,703 ccTLD: 142,145
Created Date	2004-11-08(7,415) Changes: -147 ccTLD: 3,039
Whois Servers	whois.networksolutions.com(5,593,499) Changes: -325,290 ccTLD: 35,351
Updated Date	2020-12-14(45,196) Changes: -6,761 ccTLD: 76,481
Expires Date	2022-11-08(503,691) Changes: -20,724 ccTLD: 138,189
Name Servers	NS29.1AND1.COM(5,453) Changes: -156 ccTLD: 369 NS30.1AND1.COM(5,453) Changes: -156 ccTLD: 369
Archive Date	2022-07-29

Finally, I want to buy a domain and generate email forwarding service from it. There are numerous domain registrars and web hosts which will suffice, but I prefer Cloudflare. For \$9 annually, I can own this domain and forward unlimited incoming email catch-all addresses to any external encrypted email provider, such as Proton Mail. I do not need to purchase a hosting plan from a third-party provider. For this example, I created a free Cloudflare account, which I associated with a new Proton Mail email address.

Once I was signed in to Cloudflare and presented with my account portal, I navigated to "Domain Registration" > "Register Domains". I then searched rental-bus.com and received the following result.

## Find a domain

Domain fees will be changing soon. [Click here](#) for more info.

Enter a domain name

Domain	Price
rental-bus.com	\$9.15

I purchased a domain for \$9.15 and used a masked Privacy.com card for the transaction. During the process, I was asked for my name, physical address, email address, and telephone number. These are all ICANN requirements, the entity which controls domain name registration. One could lie here, but I do not recommend it for two reasons.

- Providing false information could result in losing the domain. I have only seen this happen when domains were abused to send spam, but it could happen to us. We should obey the rules.
- Providing an alias name and non-existing email address is a sure-fire way to lose control of the domain. If you are ever required to verify ownership of the domain via email or ID, you will not be able to confirm yourself.

Therefore, let's be honest ... kind of. Any time I register a domain, I provide a shortened version of my true first and middle names as my full name. If my full name was "Michael John Bazzell", I might provide "Mich John" as my name. I have friends who call me Mike, but I have never seen them spell it. Therefore, maybe it is "Mich" in their heads. If my middle name is John and my grandmother called me Michael John often, that is my real name.

Next, they demand a physical address. I always purchase new domains while I am staying at hotels during travel. Technically, it is my home for the night. I always include the room number during my registration. I typically provide the hotel phone number as well, since domain registration is always verified over email. I provide the same Proton Mail email address which I supplied to Cloudflare as the domain registration contact. I maintain a digital copy of my hotel receipt, including my first and middle name, along with the dates of my stay and room number, in case I am ever asked to provide proof of the provided residence.

Is this overkill? Maybe. Cloudflare does not publicly share any of your registration details, and requires a court order to release that information. However, a breach or bad employee could easily eliminate all of my hard work to be as anonymous as possible. Therefore, I mask the information to a level which I feel comfortable presenting as my own.

Once I own the domain, I navigate to "Websites" and select my new domain. I then click the "Email" tab and complete the "Email Routing" requirements. At the time of this writing, the following applied. Please note that the exact wording changes rapidly at Cloudflare, so you may see some minor differences.

- Click the "Get started" button.
- Create a custom email address, such as "comms@rental-bus.com".
- Provide a destination address where your incoming email should be forwarded.
- Click "Create and continue".
- Confirm the request within your receiving email account.
- Click "Add records and enable" to apply the appropriate DNS settings.

In this scenario, any email sent to "comms@rental-bus.com" would be forwarded to the Proton Mail email address which I previously supplied. You should now click "Email" > "Email Routing" within the Cloudflare portal. Then, click the "Routes" tab and enable "Catch-all addresses". This allows any email to your new domain to be forwarded to your receiving address. If you sign up for a service using email addresses of "VoIPcall@rental-bus.com", "sales@rental-bus.com", "manager@rental-bus.com", they will all automatically forward to your reception address. This allows you to create new addresses on the fly without any email configuration. Note that these will only receive messages, you cannot send from them.

Obviously, you do not have to use Cloudflare for this. You could register a domain at any web host and pay them for email services. I prefer this route due to cost, as I own many domains which I use for specific purposes. For comparison, a domain and email hosting through Namecheap would start at \$30 annually. **You do not necessarily need a custom domain at all in order to follow the rest of this book.** If you have no plans for obtaining VoIP service, you could probably skip this step. I find it beneficial to bypass the fraud filters at most of the telephony providers, so I want everything configured before I need it. Let's proceed.

## Twilio VoIP Service

When an app or service advertises "Burner Phone", "Second Phone", "Second Line", or other enticing verbiage, they do not actually provide a telephone number or telephony services. Almost all of them rely on a VoIP service called Twilio. Even MySudo provides access through Twilio. These companies purchase numbers and service through Twilio and upsell the service to you. What if we eliminated the middle man? You could create your own Twilio account, purchase a number, and possess service without any third-party involvement. This Do-It-Yourself option is easier said than done, but a very attainable task. Many might prefer the options presented later.

The first step is to create a new account at Twilio (twilio.com) from a desktop computer. This will be the most difficult part of this entire process. You must provide a name, email address, and phone number to Twilio as part of your registration. Twilio possesses strong fraud mechanisms in order to suspend accounts which seem suspicious. During the first tests of this strategy, my accounts were immediately suspended. I had provided a vague name, burner email address, and Google Voice number while connected to a VPN. This triggered the account suspension and I was asked to respond to a support email explaining how I would be using Twilio.

This began communication with two Twilio support personnel. While talking with customer service, I was advised that the VPN IP address was most likely the reason for the suspension. After providing a business name, "better" email address, and explanation that I would be using the product for business VoIP solutions, my account was reinstated. **If you get caught within this dragnet, I discourage you to let them know you are following the protocol in this book to establish VoIP services.** Twilio does not like me or my general audience. We are small individual customers compared to big businesses.

After you create your free account, it will be severely restricted. Individual Twilio employees will analyze your registration details and decide if you can be "upgraded" into a fully-functioning account. I think you will find your account restrictions lifted within an hour if you apply the following guidelines.

- Provide your true first and middle name, especially if they are generic. In my experience, a true last name is not needed.
- If you created a custom domain, as explained in the previous chapter, provide an email address associated with this domain. Privacy-themed addresses from Proton Mail, Tutanota, or masked providers will be flagged and the account will be suspended. Gmail and other free addresses will be heavily scrutinized (or blocked entirely).
- If possible, register without protection from a VPN. I will explain VPN usage later, but this can be a trigger for all new accounts. Public Wi-Fi, such as a local library, usually works well.
- The telephone number provided could be an existing VoIP number or any landline number to which you have access. If you have an old Google Voice number, this should work well.
- A Twilio employee will likely email you and ask how you plan to use their services. Do not ignore this. You must convince them that you are worthy of paying for their product. I typically provide something similar to the following.

"I am a software developer and my boss asked me to look at the Twilio API with hopes of replacing our landlines with VoIP services. I plan to purchase a few SIP numbers and assign them to employees."

"I provide I.T. services to several companies and they are asking about VoIP services. I would like to test the Twilio API to see how that could fit into their existing systems."

"One of my customers currently uses Telnyx for VoIP services, but is unhappy with their product. They have asked me to look into the Twilio API for potential migration into your environment."

**Never use any of these paragraphs verbatim!** If we all send the same email, we will all get suspended. Take these general ideas and formulate your own reason for usage. While we are being misleading, maybe even dishonest, there is no fraud here. We will pay for the services we need. I have witnessed numerous readers' accounts become suspended when they advise that they only need a couple of numbers for personal use.

If you are required to respond to a Twilio email, and you used a custom domain with Cloudflare hosting, you have a new problem. You cannot send emails from the address you provided during registration. However, that is not required. Within your Twilio portal, navigate to "Docs and Support" > "Support Center" in the left menu and select "Ticket History". You should see a copy of any messages sent by Twilio staff. You can select the message of concern and respond directly within the portal. This will then be sent to Twilio staff from your registered email address.

Once your account is approved and you pay for the service, you will disappear into the background and you will probably never be contacted again by a Twilio employee. As long as you do not create a situation where you appear suspicious, or violate their terms of service, they should leave you alone. If Twilio demands a copy of government ID, push back. I was able to activate two accounts without ID after initial suspension. Overall, they just want paid users who do not abuse their networks.

I will now assume that you have a Twilio account created with a strong password, and that it has been upgraded by Twilio staff. The free credits in your account allows you to test many features of the service, but a \$20 deposit will be required before our account is fully usable for outside communications. The payment process is explained in a moment. I paid for mine with a masked debit card. However, I don't see a huge problem with using a real credit card. Many people will think that is reckless, and it would leave a digital trail to your true identity. This is true, but consider the following.

If you will be using VoIP numbers associated with your true identity, there will be a trail anyway. If I give a new VoIP number to my friends, family, and co-workers, it will be connected to me through usage, logs, and contact sharing. The whole point of VoIP is to have a less-invasive way to make and receive calls under your true identity. The pattern of behavior would identify you as the account holder, and that is OK.

I do not believe we need to remain completely anonymous with our VoIP provider. However, I do believe we should be anonymous with our cellular provider. If anyone investigated the VoIP account, they could probably make the association anyway based on the numbers called. Therefore, I do not see an issue with using a true credit card to pay for these services. I also don't see a problem using your true name if required. If you followed my advice in *Extreme Privacy 4th Edition* and obtained a secondary credit card in your first and middle name, even better.



Let's get back to the Twilio account. Clicking on the upper left "down arrow" should allow you to create a new account, which was once called a "project". If this option is missing, go to **<https://www.twilio.com/console/projects/summary>** and choose "Create new account". Provide a generic account name. I called mine "VoIP". This might require you to confirm a telephone number to "prove you are human". Fortunately, they accept VoIP numbers here, and I provided a Google Voice number. After confirming the number, answer the questions presented about your desired usage. The answers here have no impact on your account.

Once you have your new project created, you should see a test balance of at least \$10. It is now time to configure our VoIP telephone number. First, determine the locality of the Twilio server closest to you, based on the following configurations. I will be using the "East Coast" U.S. option, so my example server will be [phone number].sip.us1.twilio.com. The most stable option in the U.S. is "us1".

- North America Virginia: [phone number].sip.us1.twilio.com
- North America Oregon: [phone number].sip.us2.twilio.com
- Europe Dublin: [phone number].sip.ie1.twilio.com
- Europe Frankfurt: [phone number].sip.de1.twilio.com
- South America Sao Paulo: [phone number].sip.br-1.twilio.com
- Asia Pacific Singapore: [phone number].sip.sg1.twilio.com
- Asia Pacific Tokyo: [phone number].sip.jp1.twilio.com
- Asia Pacific Sydney: [phone number].sip.au1.twilio.com

If the following menu items have changed, search through their online Twilio documentation for the updates. Twilio changes their menu options often without warning or documentation. If I see drastic changes, I will update this PDF and you will be notified to download a free updated document. Let's begin.

Within the Twilio Dashboard, click "Get a Trial Number". Use the search feature to find a number within your desired area code. This will deduct \$1 from your balance. If this option is not present, click the "Develop" link in the upper left menu, then "Phone Numbers", then "Manage", then "Active Numbers", then "Buy a Number". Click "Buy" next to the desired number. My demo number is "2025551212". Proceed with the following.

- Click the "Voice" link in the left menu.
- Choose the "Manage" menu option.
- Click the "SIP Domains" option and click the "+" to create a new domain.
- Enter the assigned telephone number as the "Friendly Name", such as "2025551212".
- Enter the assigned telephone number as the "SIP URI", such as "2025551212".
- Under "Voice Authentication", click the "+" next to "Credential List".
- Enter a "Friendly" name of your number, such as "2025551212".
- Enter a "Username" of your number, such as "2025551212".



- Enter a secure password and click "Create".
- Under "SIP Registration", click the "Disabled" button to enable it.
- In the "Credentials List" drop-down, choose your telephone number.
- Click "Save".
- Navigate to <https://www.twilio.com/console/runtime/twiml-bins>.
- In the left menu click the three dots next to "Twiml Bins".
- Click "Pin to Sidebar".
- Click the "+" to create a new Twiml Bin.
- Provide a "Friendly" name of "incomingvoice".
- Place the following text in the Twiml box. Replace "2025551212" with your number.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
<Dial answerOnBridge="true">
<Sip>2025551212@2025551212.sip.us1.twilio.com</Sip></Dial>
</Response>
```

- Click "Create" and "Save".
- Click "Phone Numbers" > "Manage" > "Active Numbers" in the left menu.
- Click your telephone number.
- Under "Voice & Fax", then "A Call Comes In", choose "Twiml Bin".
- Select "incomingvoice" in the drop-down menu and click "Save".
- Click "Twiml Bins" > "My Twiml Bins" in the left menu.
- Click the plus sign to create a new bin.
- Provide a "Friendly" name of "outgoingvoice".
- Place the following text in the Twiml box.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
<Dial answerOnBridge="true" callerId=
"{{#e164}}{{From}}{{/e164}}">{{#e164}}{{To}}{{/e164}}</Dial>
</Response>
```

- Click "Create" and "Save".
- Click "Voice" > "Manage" > "SIP Domains" in the menu.
- Select your domain.
- Under "Call Control Configuration" > "A Call Comes In", change "Webhook" to "Twiml Bin" and select "outgoingvoice" in the drop-down menu.
- Click "Save".

You may have noticed a warning about an emergency call fee of \$75. This is to entice you to associate your physical home address with your account, and pay a monthly fee for the privilege. This is not required. However, any calls to 911 from this VoIP

number may generate a \$75 fee from Twilio for some reason. I would never call 911 from these numbers. If there is a true emergency, I would just use the cellular connection through the dialer app on my device. This will disclose my true cellular number to the operator, but privacy should never be a priority during an emergency.

You are now ready to receive and generate calls with your new number. You cannot do this through Twilio's website, as you will need software designed for this purpose, as explained next.

### **Twilio Sipnetic Configuration**

In previous books, I recommended a calling application called Linphone. My main motivation for this was the cross-platform availability. The same program allowed calls on Windows, macOS, Linux, Android, and iOS devices. While I still rely on Linphone for desktop calling, I no longer use it on mobile devices. The outgoing calls tend to fail and incoming calls often do not prompt the user for pickup. Today, I find Sipnetic to be much more reliable. It can be installed from Aurora Store. After installation and initial launch, you can configure it with the following steps.

- If prompted, tap "Next", "Skip", and "Yes" to exit the welcome screen.
- If prompted, allow Sipnetic to access (record) audio "While using the app" and confirm the choice. Also allow Sipnetic to "Display over other apps" by tapping it and enabling the option. Press back twice to return to Sipnetic.
- If prompted, allow Sipnetic to send notifications.
- Open the upper-left menu and select "Add new account".
- Choose "Enter Manually" and provide a "Server Name" of your number and domain, such as "2025551212.sip.us1.twilio.com".
- Click "Next" and ignore any errors about the server.
- Enter your Twilio credentials from the previous tutorials and click "Finish".

You will likely be prompted to either allow or deny Sipnetic to run in the background. If you allow this, calls will always come through even when the device is not in use. If you deny this, you will be required to open the app in order to accept an incoming call. Personally, I deny it and simply launch Sipnetic when needed. If you plan to accept unsolicited calls all day, you should allow this. There will be a very slight battery hit with this option. Next, let's modify the default configuration.

- Click on the newly created account and select "Manage accounts".
- Select the account to enter the settings menu.
- Change the display name as desired, such as "Family" or the number itself.
- Change the default transport to TLS.
- Select the option to "Use only default transport".
- Disable the "Enable ICE" option.
- Tap the check mark at the top twice.
- Tap the "Settings" item then tap the "Network" option.

- Disable "UDP", "TCP", and "Random port".
- Tap the check mark to confirm changes.

Your Sipnetic application is now configured for incoming and outgoing calls. Placing a call should be straight forward, but make note of the usage rates through Twilio. I never use these for extremely long calls. Incoming calls will ring your device as long as Sipnetic is open. If you enabled push services, incoming calls should notify you through your operating system. You will be prompted to answer the call. Sipnetic does not need to be in the foreground in the way Linphone was used. You can add as many numbers as desired within Sipnetic with the previous tutorial. Click the upper left corner in order to select your new account, or choose between multiple accounts if you add more. You should see a green check mark next to the account if the connection from Sipnetic to Twilio is successful. We can now test.

- Confirm that your Twilio account is selected within the Sipnetic application.
- Tap the orange dial pad icon in the lower left and dial a telephone number.
- Click the "phone" button to initiate a call.

If you have not upgraded your Twilio account and added funding, you should receive an automated message thanking you for using your demo account. This confirms that we can place calls to Twilio's servers, but we are far from unlimited usage to real numbers. If you can complete this test call, your configuration is complete. If you would like to remove all restrictions to make and receive calls to and from any number, you must "Upgrade" the account. The following should be conducted within Twilio.

- Return to the Dashboard in the upper left menu.
- Click the "upgrade" link and provide all requested billing details.
- Provide any credit, debit, or registered prepaid card.
- Apply \$20 to the account.

You should now have an unrestricted Twilio account which should be fully functional for voice calls. Please do not upgrade the account until you know your test calls are going through. You should also have a fully functional VoIP application which can facilitate calls. Sipnetic can be used to place a call at any time from any device. Furthermore, you can add as many numbers as you wish by repeating this process.

Before you create dozens of new numbers, let's discuss the costs. Each Twilio number withdraws \$1.15 every month from your balance. If you followed these steps, you are funded for almost three years of usage of the initial phone number. Incoming and outgoing calls cost \$0.004 per minute. During all of my testing for this tutorial so far, I spent \$1.21. There are several huge benefits with this strategy, as outlined below.

- You can now make and receive telephone calls through your mobile device without using your cellular number.
- You have more control over your number(s). You are not at the mercy of Google, and their data collection, in order to process calls.

- You can add as many numbers as desired as long as you have the funds to support them. I have five numbers through Twilio and I can access all of them through every device I own. My annual cost for this, including my usage, is about \$70. Twilio does not know my real name and only possesses a custom domain email address and Google Voice number in association to my account.
- You can port a number into Twilio. If you plan to cancel a cell phone or VoIP number, you can port it into Twilio and still have access through Sipnetic.
- You can call international numbers (at increased costs). Most VoIP providers such as Google, Twilio, and others restrict calling to nearby countries. You can enable any country in Twilio by navigating to "Programmable Voice" > "Calls" > "Geo Permissions".

Please think of this VoIP strategy as being similar to landline service. It is important to note that VoIP telephone calls and messages are not encrypted and we should expect no privacy. However, I have some isolation from my true identity. I use these numbers mostly for outgoing calls, such as calls to businesses. This strategy is an affordable option which allows telephone calls without relying on your cellular carrier-provided number. It can also be used to isolate outgoing "junk" calls which are likely to abuse your number. Twilio has the ability to see our logs, but so would any cellular carrier if we had made the calls via our official number.

The biggest feature of this process is the ability to possess affordable VoIP numbers on an un-Google'd operating system, such as GrapheneOS. We have granular control of our numbers without the need for Google's services. Any time you allow a third-party service to facilitate your calls, you are also allowing them to intercept and see your data. All of these services rely on a VoIP provider such as Twilio, so I believe we should consider creating our own solutions and eliminate any additional companies which are unnecessary.

### **Twilio SMS Messaging**

Sipnetic has no embedded voicemail or SMS/MMS text message capabilities and is only for voice calls. If you desire the ability to send SMS/MMS text messages associated with this new Twilio number, you must create an environment which can facilitate this communication. You have a few options for this, but I will present my recommended approach. The following allows you to forward any incoming SMS text messages to another telephone number, such as Google Voice, MySudo, or any other number. This is the simplest option for text message forwarding.

- Click the "Twiml Bins" option in the left menu then "My Twiml Bins".
- Click the plus to add a new bin and provide a name of "incomingsms".
- Insert the following within the Twiml field, replacing "12125551212" with your own receiving number, and click "Save".

```
<Response><Message to='+12125551212'>{{From}}:
{{Body}}</Message></Response>
```

- Click "Phone Numbers", "Manage", "Active Numbers", then select number.
- Under "Messaging", and "A Message Comes In", choose "Twiml Bin".
- Choose "incomingsms" in the field to the right and click "Save".

All incoming text messages should now forward to your other number. Note that you pay a small fee for both the incoming and the forwarding text from your Twilio balance. Advanced users may want to instantly forward any incoming SMS text messages to an email address. This requires an online web server. A shared host and any custom domain will suffice. Create a text file called `twilio.php` with the following content. Change "your@email.com" to the address where you want to receive notifications. Change "@yourdomain.com" to your actual domain name. Upload this file to your web host.

```
<?php
$to = " your@email.com ";
$subject = "Text Message from {$_REQUEST['From']} to {$_REQUEST['To']}";
$message = "{$_REQUEST['Body']}";
$headers = "From: twilio@yourdomain.com";
mail($to, $subject, $message, $headers);
```

Navigate to your Twilio dashboard and conduct the following.

- Click "Phone Numbers", "Manage", "Active Numbers", and select number.
- Under "Messaging" and "A Message Comes In", change each to "Webhook".
- Provide the full address of the PHP file you previously created within both fields. This may be similar to <https://yourdomain.com/twilio.php>.

Test your new SMS option from another number. Any incoming SMS messages to your Twilio number should now be forwarded to your email. The subject will appear as "Text Message from 2125551212 to 6185551212" and the body will contain the message sent. I prefer this option because it does not require another telephone number, such as Google Voice, in order to receive messages. When I give my car dealer this Twilio number during a maintenance visit, I receive an email when they send a text notifying me my vehicle is ready.

If you want to send SMS text messages from your Twilio number, there is a "Try it out" feature within your Twilio dashboard, but I find this process cumbersome and it relies on you to be constantly logged into Twilio. Instead, consider a Twilio "Quick Deploy" option.

First, navigate to <https://www.twilio.com/code-exchange/browser-based-sms-notifications>. Next, confirm that the "Account name" is the VoIP project which you created for this process. If you have more than one number, select the appropriate option. Finally, create a passcode which prevents random people from finding your project and sending messages. This should be a fairly secure passcode, but should also be rememberable. Click "Deploy my application" and you will be presented a URL similar to <https://sms-notifications-6431-bf4jg3.twil.io/index.html>.

Visiting this page presents a form which allows unlimited outgoing SMS text messages from your new Twilio number. Enter one or more target numbers; apply your application passcode; and write your message. Be sure to bookmark this page within your browsers in order to access it easily. If you want to send a response to a received message, you can open your new Twilio page and send it from there.

**To be transparent, I do not do this.** It is simply too much effort. Also, Twilio has an unknown threshold regarding outgoing text messages. If you surpass it, they will demand either a SSN or EIN issued by the IRS in order to continue service. This is to combat SMS spam. I want no part of that. I view these VoIP numbers as a way to make and receive telephone calls, and receive text messages. If you are looking for a way to send unlimited messages to others, consider the secure encrypted options presented later in the book.

### Twilio Voicemail Configuration

Next, consider voicemail. Some may prefer to have no option to leave a voice message. The instructions up to this point will either ring your Sipnetic application for 30 seconds and then hang up, or simply terminate the call right away if Sipnetic is not open and connected. I prefer this for some numbers, as I do not want the caller to be able to record a message. However, we can enable voicemail, tell Twilio to record the message, save it to their servers, and email us a link of the recording. Conduct the following within the Twilio Dashboard.

- Navigate to <https://www.twilio.com/labs/twimlets/my/> to access Twimlets.
- Choose "Voicemail" then "Create New Twimlet".
- Provide your desired email address to receive voicemail notification.
- Provide your desired outgoing greeting.
- Choose "True" to have the messages transcribed to text or "False" to avoid transcription. Note that transcriptions add an extra cost and do not impact the ability to hear the voice messages. I do not transcribe them for privacy reasons.
- Click "Save URL" then provide a nickname of "voicemail".
- Copy the URL, similar to "<http://twimlets.com/AC5b84e8/voicemail>".
- Click "Twiml Bins" in the left menu and select "incomingvoice".
- Replace the current text with the following.

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <Dial answerOnBridge="true" timeout="30"
    action="http://twimlets.com/AC5b84e8/voicemail">
    <Sip>2125551212@2125551212.sip.us1.twilio.com</Sip>
  </Dial>
</Response>
```

- Replace "<http://twimlets.com/AC5b84e8/voicemail>" with your URL.
- Replace "2125551212" with your own number.

- Replace "us1" with your own server location if necessary.
- Click "Save" and test the service.

If your Sipnetic application is open and connected, an incoming call should ring for 30 seconds. If you do not pick up the call in that time, the voicemail system presents a generic greeting and allows the caller to record a message. If Sipnetic is closed or not connected to Twilio, the greeting is presented right away. If a caller leaves a voicemail, you will receive an email at the address provided which includes a link to hear the recorded MP3 file. This recording can also be accessed by navigating to "Voice" > "Overview" in your Twilio Dashboard.

Similar to Google Voice, you can delete the recorded file from this menu. This file is not secure or private. It is very similar to the way a traditional cellular provider or Google Voice would store voicemails available to your device. If you have no devices connected to your Twilio account which are ready to receive a call when a call comes in, expect to see error messages within the Twilio "Monitor" menu. These are to notify you that your phone system could not receive the call and can be ignored.

Before you commit to voicemail transcription, consider my thoughts on Twilio account sanitization, which are presented in the next section. If desired, disable the "Daily Calls Log Archives" logging feature within Twilio at "Voice" > "Settings" > "Log Archives". This does not stop Twilio from storing VoIP call metadata, but it does eliminate a small layer of internal logging.

Keep in mind that additional numbers will extract funds faster. I only recommend additional numbers if you understand the reasons which you need them. Repeat the previous steps for each number needed. While writing this update, I configured a toll-free number. The monthly fee for this number is \$2.00 (twice the price of a standard number), but it presents a more professional appearance. I have also witnessed toll-free numbers behave differently when used as number verification. One of my banks absolutely refused any VoIP number as my required 2FA authorization number. However, providing a VoIP toll-free number passed the scrutiny. When I attempted this on PayPal, a toll-free number was absolutely refused. There seems to be no standards with this. Testing different options might lead you to your own best option.

You can now choose between multiple different numbers within your Sipnetic application. Whichever is chosen as default allows outgoing calls to be completed from that number. Incoming calls to any numbers will ring the app and allow connection regardless of the default account. Incoming text messages will be stored at the Twilio Dashboard and voicemail will be transcribed and sent to your email address. You can replicate this for unlimited numbers, as long as you have funding to support them.

### **Twilio Account Sanitization**

If you use any manual SMS/MMS messaging option, message metadata and content remain on Twilio's servers, and could be accessed by employees. Every voicemail you receive also stays present on their servers as an MP3 file, which can be accessed via



direct URL without any credentials. Let's identify ways to remove this data, beginning with stored text messages.

- Navigate to <https://console.twilio.com>.
- Make note of the "Account SID" and "Account Token".
- Click on "Messaging" then "Overview" in the left menu.
- Open any "Recent Message" by clicking the date and note the "Message SID".

You can now open Terminal within any Linux or Apple system and issue a command to delete each message. If your "Account SID" was 11, "Account Token" was 22, and "Message SID" was 33, the command would be as follows.

```
curl -X DELETE https://api.twilio.com/2010-04-01/Accounts/11/Messages/33.json \
  -d "Body=" \
  -u 11:22
```

This can be quite annoying if you need to purge hundreds of messages. Voicemail and call log deletion is more straightforward within the website. The following steps allow you to remove this data from your console.

- Navigate to <https://www.twilio.com/console/voice/dashboard>.
- Open any log entry which has an arrow icon under "Recording".
- Click "Delete this call log" and confirm.
- If desired, delete individual call logs from this location.

Twilio stores 13 months of call log history by default. If you possess numerous recordings which need removed, you can use the bulk deletion tool with the following directions.

- Go to <https://www.twilio.com/console/voice/recordings/recording-logs>.
- Click "Select" and then "Select All".
- Click "Actions", "Delete Recordings", then confirm.

If you have enabled the call transcription service, you may wish to remove all voicemail text transcriptions stored within your account.

- Click "Monitor", "Logs", then "Call Transcriptions" in the left menu.
- Open each transcription and click "Delete this transcription".

While writing this section, I realized that my data had not been sanitized for a long time. My Twilio dashboard possessed voicemails and text transcriptions about my health, family, friends, and work. I spent an hour cleaning all of it, then disabled transcriptions using the previous tutorials. It saves me \$0.05 per call and eliminates one more place where sensitive information could be stored. We can also disable some logging by Twilio with the following modification.

- Click "Voice", "Settings", and "General" in the left menu.
- Disable "Request Inspector" and click "Save".

If you are interested in a guide to Twilio sanitization from a desktop computer, please visit my guide at <https://inteltechniques.com/voip.twilio.cli.html>. I use this every day to clean up my tracks.

All of this logging may seem invasive. It is, but it is not unique to Twilio. Twilio is doing nothing more than every other telephony provider including cellular and landline telephone companies. Fortunately, we have some control of how the data is stored. However, I do not want to present false expectations here. While Twilio may appear to have deleted your call logs, voicemails, messages, and transcriptions, they are all likely still stored somewhere within their system. Our only goal is to remove the data from within our dashboard. Never expect any level of privacy when it comes to traditional phone calls and messages. VoIP services should never be used for sensitive communication. Assume there is a log of everything which will be stored forever.

### **Telnyx VoIP Service**

In past writings, I highly recommended a Twilio alternative called Telnyx. At the time, they were less scrutinous of new accounts and encouraged people to try their services. Today, I urge caution before proceeding. This is due to several issues.

- Telnyx only provides accounts to confirmed businesses. However, your new custom domain email address may suffice.
- Telnyx no longer provides actual customer support. Support tickets only receive canned responses, and the request is eventually closed without a solution. Calls to their support line inform you to send an email, which is never answered. You will need to do your own troubleshooting if required.
- Telnyx does not provide voicemail services.
- Telnyx does not allow you to delete your user logs or text messages from their system in the way Twilio does.
- Telnyx suspends paid accounts if their automated fraud system detects unusual activity. I have experienced this myself when an unused account appeared suspicious to them. When I questioned about this practice, I was ignored.

However, I know of many people who prefer Telnyx over Twilio. Their monthly number fee is slightly less and redundancy is always a good thing. If the Twilio tutorial did not generate the usage you desire, possibly due to a suspended account, you might consider Telnyx (<https://refer.telnyx.com/refer/zrfmo>). This VoIP provider replicates the service provided by Twilio, but their setup process is much easier. Now that you have an understanding of our Twilio strategy, I will abbreviate the steps here for Telnyx.

- Create a free account at <https://refer.telnyx.com/refer/zrfmo>. This specific URL provides \$20 in free credits which can be used right away.

- Provide a custom domain email address, which was previously explained.
- If prompted for purpose, choose "SIP Trunking".

You should now be logged into the Telnyx portal. You can now create your first connection and purchase a telephone number.

- Click "Voice" then "SIP Trunking" from the side menu.
- Click the "+ Add SIP Connection" button.
- Enter the name you wish to have for your connection (I chose "VoIP").
- Click "Create Sip Connection".
- Enable "Credentials" as the "Connection Type".
- Copy the username and password automatically generated.
- Click "Save and finish editing".
- Click "Numbers", "My Numbers", and "Search & Buy Numbers".
- Enter a location and click "Search Numbers".
- Choose a number and click "Add to Cart".
- Click the "Cart" in the upper right.
- Under "Connection or Application", select your connection (mine was previously created as "VoIP").
- Purchase the number using your free credits by clicking "Place Order".
- Click "Voice", "Outbound Voice Profiles" then "Add new profile".
- Provide the name of "outgoingvoice" and click "Create".
- Click "Outbound Voice Profiles" then the "Edit" icon next to "outgoingvoice".
- Select your connection (VoIP) and click "Add Connection/Apps to Profile".
- Click "Voice", "Sip Trunking", "SIP Connections" then "Outbound Options" to the right of the connection.
- Enter your new phone number in "Caller ID Override", then click "Save".

### **Telnyx Sipnetic Configuration**

- If prompted, tap "Next", "Skip", and "Yes" to exit the welcome screen.
- If prompted, allow Sipnetic to access (record) audio "While using the app" and confirm the choice. Also allow Sipnetic to "Display over other apps" by tapping it and enabling the option. Press back twice to return to Sipnetic.
- If prompted, allow Sipnetic to send notifications.
- Open the upper-left menu and select "Add new account".
- Choose "Enter Manually" and provide a "Server Name" of "sip.telnyx.com".
- Click "Next" and ignore any errors about the server.
- Enter your credentials provided by Telnyx and click "Finish".

You will likely be prompted to either allow or deny Sipnetic to run in the background. If you allow this, calls will always come through even when the device is not in use. If you deny this, you will be required to open the app in order to accept an incoming call.

Personally, I deny it and simply launch Sipnetic when needed. If you plan to accept unsolicited calls all day, you should allow this. There will be a very slight battery hit with this option. Next, let's modify the default configuration.

- Click on the newly created account and select "Manage accounts".
- Select the account to enter the settings menu.
- Change the display name as desired, such as "Family" or the number itself.
- Change the default transport to "TCP".
- Select the option to "Use only default transport".
- Disable the "Enable ICE" option.
- Tap the check mark at the top twice.
- Tap the "Settings" item then tap the "Network" option.
- Enable "TCP".
- Disable "Random port".
- Tap the check mark to confirm changes.

Your Sipnetic application can now make and receive calls without adding any funds. This is unique to Telnyx. If you want to commit to Telnyx as your VoIP provider, be sure to add \$20 in new funds to your account in order to prevent termination of the trial. This provides enough credits (\$40) to provide VoIP service for over three years, including a single number and usage.

Telnyx does not offer native SMS forwarding to their web portal or another number. The only option is self-hosting a forwarder to an email address as we did with Twilio. If you have your own domain and a shared web host, create a text file titled `telnyx.php` with the following content. Change "your@email.com" to the address where you want to receive notifications. Change "@yourdomain.com" to your actual domain name.

```
<?php
$to = "your@email.com ";
$subject = "Text Message from {$_REQUEST['From']} to {$_REQUEST['To']}";
$message = "{$_REQUEST['Body']}";
$headers = "From: telnyx@yourdomain.com ";
mail($to, $subject, $message, $headers);
```

Upload the file to your web host. Afterward, your specific URL may be similar to <https://yourdomain.com/telnyx.php>. Within Telnyx, conduct the following.

- Click "Messaging", "Programmable Messaging", then "Add New Profile".
- Provide a name of "sms" and select "Twexit API".
- In both "webhook" fields, enter the URL of the PHP file previously created.
- Click "Save".
- Click "Numbers" then "My Numbers" within the left menu.
- Within your number entry, select "sms" in the "Messaging profile" field.
- Confirm the rate notice if prompted.

Incoming text messages should now be forwarded to your email address. The subject will identify the sender and recipient while the message body will display the text message. This method prevents Telnyx from storing all of your incoming messages (content) on their own server in the way that Twilio does, but they still maintain a permanent log. They would still have the ability to intercept and see the contents, but that is unlikely. Once the message is routed to your email, you should be the only host of the content.

If you want to send a text from your new Telnyx number, click "Messaging" > "Programmable Messaging" > "Learn & Build" > "Send & Receive a Message". You can use the online form to send a SMS text message to any number. You can also use the commands provided on that page to send messages from within Terminal. Similar to Twilio, I do not use this feature. I never use a VoIP number for back-and-forth conversations. I only need to receive the occasional confirmation text message, which forwards to my email from both providers.

You can customize the caller ID name displayed during your outgoing calls within the Telnyx portal. Click "Numbers" > "My Numbers" from the menu and then "Caller ID/CNAM Listing" under the services area of your chosen number. Enable the "CNAM Listing" and "Caller ID Name" options, then enter any name desired. It may take a week to take effect. Be sure to enable two-factor authentication (2FA) through "My Account" in the "Security" section.

While this configuration is simpler than Twilio, it has less features. However, there are also benefits which are not available with Twilio. Consider the following.

- With Twilio, unanswered calls went directly to voicemail, and messages were transcribed and emailed to me. With Telnyx, unanswered calls disconnect after about 30 seconds. There is a voicemail option, but it requires a paid third-party service. If you want voicemail and transcription, Twilio is best.
- Twilio allows incoming text messages to be natively delivered directly to your dashboard or forwarded to any other number. Telnyx requires you to host your own message forwarding server for this to work. If you need the number to support incoming SMS text without third-party services, then Twilio is the appropriate option. If you have your own website, replicating this is fairly easy.
- Twilio possesses numerous fraud triggers which can impact our usage. Many readers report difficulties simply creating an account and being allowed access. Telnyx provides immediate access upon registration of a "business" email address. However, I have witnessed Telnyx suspend accounts created using free email domains behind a VPN. Always provide an email address associated with a custom domain while connected to public Wi-Fi in order to present the highest chance of obtaining a new account. Since you will be using this service to make and receive telephone calls associated with your real name, I see very little reason to attempt registration with an alias.
- The pricing and overall call quality for Telnyx and Twilio is almost identical.

I currently maintain numbers through both services and configure each into Sipnetic. If I were forced to rely on only one service, it would be Twilio due to the voicemail options and overall stability. If you have a Telnix account and it works for you, great. However, I encourage others to focus their efforts on Twilio. I currently maintain several numbers through Twilio and configure each into Sipnetic. Incoming SMS text messages are forwarded to my email account via my website. The service is not perfect. Twilio often changes their settings without notice and support is not always responsive. In a moment, I provide further summary of detailed usage of all services for myself and clients.

## **VoIP Issues**

During testing, I attempted to replicate these services with Bandwidth LLC and VoIP.ms. I do not recommend either of these companies. Bandwidth refused my numerous requests for service and VoIP.ms demanded unredacted copies of my driver's license before an account would be confirmed. When I refused, they closed my account which had a funded balance. While Twilio had their own roadblocks during account creation, they were the first VoIP service which actually provided me service. Anticipate fraud-related hurdles, but know that you can break through the temporary annoyances. Many people ask about services such as JMP.chat. JMP also uses Twilio numbers, but charges \$3 monthly (three times the cost). I see no reason to pay that to a middle man when you could buy your own numbers for less.

VoIP solutions often have limitations over traditional cellular communications. Twilio, and any services which rely on Twilio, do not always support "short codes". These are abbreviated phone numbers that are usually 5 or 6 digits in length. They are commonly used to send SMS and MMS messages with verification codes for account access. I think of these numbers as landline replacements which allow me to send and receive voice calls and personal texts. I maintain a single Google Voice account which can receive short codes when needed.

With GrapheneOS, or any other Android device, Sipnetic stays open after initial launch and "listens" for incoming calls while inactive. This means you must launch the Sipnetic application once after each reboot in order to accept incoming calls.

I have witnessed temporary number suspension from Twilio if Sipnetic is misconfigured. Since Sipnetic stays open and connected at all times, it may be synchronizing with Twilio servers too often with unique data. Disabling "Random Port" and confirming "TLS" as previously explained should help avoid this error. If you continue to receive warnings about connections, identify the issue. Spend the time to correct the issue once for future usage without disruptions.

By default, there is no name associated with the caller ID when you place a call from your Twilio number(s). This may be desired by some, but could be a disinformation campaign for others. On one of my Twilio numbers which I use for personal calls in my true identity, I attached my name to the caller ID. This way, my name appears as the caller on the screen of my bank or credit card company when I call. It adds an extra layer of assurance. On another number, which I use with my alias name, I prefer

that name to display as the caller. This also adds credibility to my call as an alias. Twilio requires you to contact their support in order to request these modifications.

Overall, I view this method as a simple and affordable phone line which provides unlimited numbers at my disposal. I can place calls from my mobile device when needed without exposing my true cellular number. I can accept incoming calls on my GrapheneOS device as if it was a traditional landline telephone. The person on the other end does not know I am using VoIP instead of a standard phone line. VoIP calls while possessing a stable internet connection can be much more reliable than cellular calls with a weak signal.

### **MySudo Configuration**

Many of my clients currently use the VoIP service MySudo ([mysudo.com](https://mysudo.com)) for most non-secure communications, such as incoming and outgoing telephone calls. This app provides up to nine profiles, and each profile possesses a unique telephone number, email address, and contact list. This service allows me to possess multiple phone numbers on one device, and each can be used for incoming and outgoing calls and text messages, all without the need to configure VoIP numbers and services. It requires a traditional iPhone or Android device, but can also work on our GrapheneOS device through Aurora Store. This requires more explanation.

Traditional iPhone and Android users can download MySudo, generate a new account, and pay for premium features from their device. The Google Play Store or Apple App Store facilitates the registration and payment conveniently. We have a different issue. Our secure GrapheneOS device does not have a functioning version of the Google Play Store with an associated Google account (nor should it). This prevents the ability to pay for the service through Google.

GrapheneOS users have only one reliable option. We must maintain a second phone for registration and activation of a MySudo account. I don't find this to be a huge hassle. I have an old iPhone SE which I used several years ago. It has an Apple ID registered to an alias name. I maintain a balance within the app store paid via a gift card. Once a year, I turn on my iPhone; open MySudo; and renew my account. This purchase synchronizes to the MySudo installation within GrapheneOS immediately. Neither Apple nor MySudo knows my name. You could also replicate this with a stock Android device associated with a Google account.

MySudo does not need your name, email address, or telephone number. The installation is unique to your hardware. MySudo only knows you by this "fingerprint", which has no association to your true identity. The advantage of MySudo is convenience. The work is done for you, and the application can be much more reliable. The disadvantage is that it only works within a mobile environment.

If you choose to place MySudo on your GrapheneOS device, install it from Aurora Store. Open MySudo on whatever device maintains an active account and choose the export feature from the settings. On your GrapheneOS device, choose the import option. The instructions will ask you to capture the QR code presented on one device



with the camera of the other. During my testing, I had to attempt the import/export option up to three times before it would take. This pairing should only need done once.

If you possess Google's push services on your GrapheneOS device, you will receive push notifications within MySudo. You can answer calls in real-time and receive notifications of incoming text messages. Everything works almost identically to the traditional versions. If you do not have push services installed, the app must be open to receive incoming content. You will also need to pull down to refresh the screen to see new content. Most clients using MySudo have push services, and none of this is an issue.

I currently use MySudo within my GrapheneOS device. I have the Sudo Max plan which gives me nine profiles. Each profile has its own VoIP telephone number and MySudo email address. Outgoing calls and texts are reliable, but incoming calls and texts are missed since I do not have push services installed. I simply launch the MySudo app occasionally throughout the day to identify any missed communications.

Since most of my clients have push services enabled, MySudo works for them as it would anyone else. Incoming calls ring the device and can be answered from the lock screen. Incoming text messages present a notification and alert if desired. This is a strong benefit of GrapheneOS's sandboxed push services.

Combined with my numbers from Twilio along with Sipnetic, I have over a dozen numbers at my disposal. I have never found myself without a working way to make and receive calls and texts. I remind you again that redundancy is key to this lifestyle. In the interest of full disclosure, I served as an advisor to Anonymo Labs (the maker of MySudo) for two years during the early development of this service, and I possess shares of the company.

## **Number Porting**

Now that you have a new mobile device with new anonymous service, you likely need to make a decision about your previous device and service. You could cancel the account and lose the number forever; keep the plan and check the old device occasionally for missed calls and messages; or port your old number to a VoIP account. I prefer porting over all other options, but let me explain why before providing instructions. If your old device is out of contract, you have the right to discontinue service. If it possessed a prepaid cellular account, you can suspend the service and simply stop using that plan. Most readers likely possessed a device with a contract through a traditional carrier. If you are still under contract, it may be more affordable to keep the plan until it expires. If it is a newer contract, it may be more affordable to pay an early termination fee. Regardless, at some point the plan will be discontinued. When that happens, you lose all access to that number. Any incoming calls and messages will be lost, and you will not be able to use that number for any sort of verification process, such as calling your bank to make changes to an account.

I do not believe you should ever lose a telephone number that has ever been important to you. When you change your number and start providing a VoIP number, such as a Twilio, MySudo, or Google Voice number, it is unlikely you will remember to contact everyone who has your old number. This can lead to missed calls from old friends or lost text message reminders from services you forgot to notify. Worse, someone will eventually be assigned your old telephone number if you do not maintain it. That stranger will start receiving calls and messages intended for you. Think about any time you obtained a new telephone number. You likely received messages meant for the previous owner. A mischievous person could have some fun with that.

I will assume that you are ready to port over your old number to a new permanent holding place. If you are out of contract, you are in the clear. If a contract exists, you will be held responsible for any early termination fee. I have found that notifying your current carrier and providing a new physical address as your new home which cannot receive their service is sufficient for waiving any fees. I have yet to find a carrier which can provide service to the following address, in case you find this information to be helpful.

10150 32nd Avenue NW, Mohall, ND 58761

The most important first step is to not cancel your service with your old carrier. If you do this, the number is lost and you have no way to port it over. Your account must be active and in good standing in order to port your number to another service. Once you successfully port the number over, that action will terminate the original account. This may make more sense after we walk through the process together. In the following scenario, you have recently purchased a new device, executed new prepaid service, and you still possess your old phone with the original service still active.

As you may recall, I am not a fan of Google products from a privacy perspective. However, Google Voice is our current easiest and most affordable option for porting numbers. Once we have the process in place, there will be no need to log in to the Google account, and you will never do so from your new clean device. Google will receive information about your communications through their service, but I do not see it as any worse than your previous telephone carrier possessing the same data. I present an alternative option in a moment.

The first consideration is to identify which Google account to use for the porting. If you have never had a Google account, you have no choice but to create a new one. Many people may think that a new account should be mandatory for this procedure, but I have a different view. Google can be cautious when it comes to new accounts. If you create an account behind a VPN using a burner email address, Google might find this suspicious and suspend the account until you upload government identification proving your identity. I find this invasive. I respect their need to block usage from spammers, scammers, and other crooks, but I do not want to have my own account suspended. If you already have a Google account established in your true name, and your old phone was also established in your true name, I see no reason why you should not pair these together.

Remember, our goal is to configure a system to receive calls and messages from a number that was already associated with your true identity. Connecting this to a Google account under your true identity does not gain or lose much privacy at this point. I would rather attach your old number to an aged Google account that has very little risk of being suspended due to questionable activity than to connect it to a brand-new account which will be scrutinized by Google.

If you have an old Google account in your name, I suggest using that. If you have no account, I would create an account in your true name. This may sound ridiculous from a privacy perspective, but if it gets suspended, you have a much better chance unlocking it when you are the person with whom it is registered. It will receive extremely minimal use, and Google will collect very little information from it. Let's get started.

- Find your billing account information from your current service provider, such as your account number and PIN. You need this information to complete your port request.
- Within a browser while protected by your VPN, navigate to [voice.google.com](https://voice.google.com).
- Sign in with your Google account credentials.
- If you have not used Google Voice on your account before, set up a new Google Voice account. You'll be prompted to pick a new number, but your ported number will soon replace it, so it will not matter what that number is. You can use your old cell number as your verification number, as it is still active on the old device.
- At the top right, click "Settings".
- Click "Transfer" under your number.
- Next to your current number, click "Change / Port".
- Select "I want to use my mobile number". Follow the onscreen instructions to set up your new number and pay. Google will charge a \$10 fee for the porting. You might be charged a \$20 fee to port your mobile number to Google Voice from some mobile service providers, such as Verizon or AT&T. Since your account is already in your true name, I provide a traditional credit card during purchase.
- Continuously check the status of your number porting. Numbers typically take from 48 to 96 hours to port.
- Do not cancel your phone plan until Google notifies you the port is complete. To verify the port, they will call your phone with a code. After the port is finished, your service provider will cancel your phone service.
- If you have multiple numbers on the original account, check with the service provider first to find out about their policies. If you want to keep the plan and get a new mobile number, confirm that with the service provider.

Once you see your old number which was previously attached to your cellular telephone appear as your new number in the Google Voice account, the porting is complete. Test this by completing the following steps.

- While logged into your Google account, navigate to mail.google.com.
- Navigate to www.callmylostphone.com and enter your telephone number.
- On the Gmail screen, you should see an incoming call.

There is no need to answer this call, you just want to make sure that the number can receive calls through Google Voice. You are finished with this step. If anyone from your past calls your old number, you have a way to receive notification of the call. This applies to text messages as well. You have control of the number. If you need to make a call from that number, such as to prove your identity to a bank, you can make calls from the Gmail or Voice pages while logged into the Google account in a web browser.

Having the ability to occasionally check the Google account may be all you need. Personally, I do not like logging in to Google products, so I take advantage of their forwarding options, as explained soon. It should be noted that Twilio and MySudo also offer number porting options into their network. I believe Google Voice is still the best option which will not generate monthly fees for access to the number. It also allows us strong security with two-factor authentication. However, hosting your own ported number has some privacy advantages, as explained next.

### **Porting Into Twilio**

In 2021, I needed to port two numbers. One was with T-Mobile and the other with MySudo. The T-Mobile number was provided through the prepaid provider Mint Mobile. I wanted to cancel my current account with them and start over with a new SIM. This is always a great opportunity to port a "real" number into a VoIP provider. Many companies which scrutinize VoIP numbers will often allow a number which was originally assigned to a traditional carrier, even if the number has since been ported to a VoIP provider. The following documents the entire process.

On September 10th, 2021, I navigated to <https://www.mintmobile.com/chat> and began a text support session. I requested the "account number" and "PIN" associated with my account. This was immediately met with skepticism and a demand to know why I needed these details. I advised I was moving to another country where there was no T-Mobile coverage and I wanted to port the number to a new provider. The representative confirmed my account details (name and number) and sent a temporary verification code via text message to the cellular number. It is vital that you either have cell coverage or calling via Wi-Fi enabled during this process. After confirming the verification code to the representative, I was provided my account number and PIN (last four digits of the cell number). The account number is required for porting.

On September 11th, 2021, I began the porting process with Twilio. I completed the "Letter of Authorization" form which is available on their site. This is where I encountered my first issue. My Twilio account details include my real name in order to prevent account suspension. My Mint Mobile account had alias details. If these two sources do not match, the request will be denied. I modified the Mint Mobile account details to reflect my first initial as the first name and a misspelling of my last name (Bazel). Since I was no longer using this account, I saw this as acceptable. On the

"Letter of Authorization" I made sure my name on the form was identical to the Mint Mobile details, but included my name at bottom as "M. Bazzell". This is close enough for porting. I provided a random hotel address in NYC and signed the form. I included a screen capture of the Mint Mobile account displaying my new name details and submitted all information through the Twilio porting website.

While I waited for a response, I associated this real cellular number with a new Google, Gmail, and Google Voice account. Since a real cellular number is required to generate a new Google Voice number, and I was going to port this number into VoIP anyway, I figured I may as well collect yet another voice number. I could have also associated this number with any other online account which required a "real" number, such as a bank, social network, or credit card.

On September 14th, 2021, Twilio confirmed the porting request and submitted it to the carrier (Mint Mobile). On September 16th, 2021, Twilio confirmed the porting request was received by Mint Mobile, and scheduled the final port for September 29th, 2021. On that date, the port completed. I was unable to log into that account via Mint Mobile, and the number was available within my Twilio dashboard. I used the previous tutorials to set up voice and text communication.

This Mint Mobile (T-Mobile) cellular number was now a Twilio VoIP number. However, many online services will still assume it is a true cellular number since it is within a block of numbers originally assigned to T-Mobile. I can continue to associate this number with various accounts, and those services will think I am providing a true cell number. This will not last forever. Various carrier identification services will eventually update their records. However, services locked in before that date should continue to function. This is why I always associate my true cellular number with various accounts during the interval between the original porting request and the final porting process.

The process for MySudo was much simpler and faster. I wanted to port a number I had been using with MySudo into Twilio so I could use it with the VoIPSuite application. I emailed Twilio and explained that I possessed a Twilio number through the service provider MySudo and wanted that number ported into my own Twilio account. I had to copy MySudo support in the email and they had to confirm the request with Twilio. Since the number was not leaving the Twilio network (MySudo numbers are provided by Twilio), the entire process was completed in a few days.

I must now pay \$1 per month for each of these numbers, but I am in control of them. I remind readers that I am extreme in my methods, and this is not appropriate for everyone. I like to test the limits of various methods, including number porting, mostly to learn of any pitfalls my clients may face. If you have the need for several numbers, it may be appropriate for you to port any cellular numbers you will be losing. For most, this is overkill.

Many readers have attempted to port numbers out of prepaid providers during a free trial period. This almost never works. When you contact Mint Mobile support and your number is still within a free trial period, you do not "own" the number. The

representative is very unlikely to give you the account number and allow porting out of that number.

## **Number Forwarding**

Over the years, I have accumulated many numbers from Google Voice. Some of these are heavily associated with my true name. As an example, I used a Google Voice number when I worked as a Detective at a police department. We were all required to disclose our cell numbers on a callout list, and I only provided a Google Voice account. To this day, I hear from former colleagues through that number. Many of them assume it is my cell number, and I have no need to correct them. While I have moved all of the people with whom I continuously communicate over to better options, this Google number still receives a lot of activity. The following explains how I interact with these numbers without using official Google websites or apps.

First, let's assume that you have either a Twilio or MySudo VoIP number of 202-555-1111 and email address of VoIP@protonmail.com. Any calls to that number will ring your phone through your VoIP provider and incoming emails will be received within your Proton Mail inbox. Your telephone carrier and manufacturer will not know of these calls or messages. Next, conduct the following.

- In a browser, navigate to [voice.google.com](https://voice.google.com) and select "Settings". Your Google Voice number could be the old cell number which you ported into Google.
- The "Linked Numbers" section should either be blank or possess the same number as your previous cell number. Remove any numbers within this block.
- Add a "New linked number" of your VoIP number for forwarding (202-555-1111).
- Confirm the code sent via SMS text to that number.
- In the "Messages" section, ensure that messages are forwarded to the Gmail account for this profile.
- In the "Calls" section, ensure that call forwarding is enabled.
- In the "Calls" section, ensure that "Get email alerts for missed calls" is enabled.
- In the "Voicemail" section, ensure that "Get voicemail via email" is enabled.

Let's pause and think about what is in place now. If anyone calls your old cell number, which was ported to Google Voice, the call is routed through Google Voice and then to your VoIP number. Your VoIP number will ring as normal and you can accept the call. The caller ID will show the number calling you. If you decline the call, the caller will be sent to your VoIP voicemail (if available). If you simply do not answer, it will be sent to the Google Voice voicemail. If the caller leaves a voice message within your Google Voice account, it will forward to your Gmail (which we will soon forward to Proton Mail). If someone sends you a SMS text message to this old number, it will also be received in the email account. Let's forward those messages in order to prevent checking the Gmail account.



- Navigate to gmail.com while logged into your account.
- Click the gear icon on the right and select "Settings".
- Click the "Forwarding and POP/IMAP" option in the upper menu.
- Click "Add a Forwarding Address" and enter the desired email address.
- Google will send a confirmation email to your account.
- You should now have the option to select "Forward a copy of incoming mail to" and choose your email address in the drop-down menu. Choose "Delete Gmail's copy" and save your changes.

Now, when someone leaves you a voicemail or sends you a text message to the Google Voice number, it will appear in your primary email and Google will delete the original after 30 days. You can now receive calls, voicemails, and text messages from your old number within your VoIP and email strategies without ever logging in to Google again. You can also respond to text messages via your email address and the recipient will only see the previous cellular number that is now assigned to Google Voice. I do not recommend this since the message is sent on behalf of Google. It is vital to test all of these options before relying on them. If you have VoIP, test all calling and texting options and make sure everything appears as desired. If you do not have a VoIP solution, let's repeat the entire process with alternative options.

- In your web browser, navigate to voice.google.com, click on the left menu, and select "Settings". Your Google Voice number should be the old cell number which you ported into Google.
- The "Linked Devices" section should either be blank or possess the same number as your previous cell number. Remove any numbers within this block by clicking the "X" next to each.
- In the "Messages" section, ensure that messages are forwarded to the Gmail account for this profile.
- In the "Calls" section, ensure that "Get email alerts for missed calls" is enabled.
- In the "Voicemail" section, ensure that "Get voicemail via email" is enabled.

If anyone calls your old number within this configuration, the call is routed through Google Voice and then immediately to voicemail (unless you are logged into Google Voice via web browser). If the caller leaves a message, your email account will receive the audio and text version of the call. If someone sends you a SMS text message to this old number, it will be received in the email account as well. Now, let's forward those messages in order to prevent checking the Gmail account at all, similar to the previous steps.

- Navigate to gmail.com while logged into the account associated with the old number.
- Click the gear icon on the right and select "Settings".
- Click the "Forwarding and POP/IMAP" option in the upper menu.
- Click "Add a Forwarding Address" and enter your email address.
- Google will send a confirmation email to your account.



- You should now have the option to select "Forward a copy of incoming mail to" and choose your email address in the drop-down menu. Choose "Delete Gmail's copy" and save your changes.

Now, when someone leaves you a voicemail or sends you a text message, it will appear in your email account and Google will delete the original email after 30 days (the text messages must be manually removed). You cannot receive calls, but will be notified of voicemails and text messages from your old number without ever logging in to Google again. You can also respond to text messages via your email address and the recipient will only see the previous cellular number that is now assigned to Google Voice. Again, this should be tested before actual use.

I have replicated this process across many of my old Google Voice numbers. This may seem sloppy, as Google now knows I am the owner of all of the accounts. My stance on this is that it likely does not matter. Google probably already knows. Their heavy use of browser fingerprinting, analytics, and IP documentation allows them to know when people use multiple accounts. Since I no longer have these numbers as part of my normal usage, I consider them all "burned" and only wish to have the ability to receive any notifications. Note that Google allows any VoIP number to be connected with only one Google account. We can no longer forward multiple numbers to a single VoIP number. We can also no longer forward SMS text messages to other VoIP numbers, but I never used this feature anyway.

If you call any of my old numbers, my primary device receives the call through various VoIP numbers. If you send a text to any of my old numbers, they are received in my email inbox. I never use these Google accounts to make any outgoing calls or send texts. These are only used for incoming content from people who do not know my true new number(s). This presents a small annoyance with this plan. You can only call out from your old Google Voice numbers if you log in to the corresponding Google account. I try to avoid this unless the caller ID on the other end needs to be the old Google Voice number.

There are a few reasons you may need to do this. Imagine that you contact your credit card company in reference to your account. The cellular telephone number that they have on file is your previous Google Voice account. For security purposes, they mandate that you contact them from a known number to protect your account. You could call from the Google Voice dashboard and the number would be sent through via caller ID. If you do need this outgoing call feature, consider associating a dedicated browser for this purpose. Brave is based on Chromium (Chrome) and works well with Google Voice. I prefer to eliminate association with any Google accounts within my primary browser.

**VoIPSuite** ([inteltechniques.com/VoIP.suite.html](http://inteltechniques.com/VoIP.suite.html))

As I was finishing this chapter, I was notified by the creator of VoIPSuite that an Android version was available within F-Droid. VoIPSuite is an open-source software application which can easily facilitate incoming and outgoing text messages via a Twilio

account. I have successfully used the web-based version in the past, but have yet to test the Android release. If you rely on Twilio for VoIP service and need a way to send and receive text messages, please check out VoIPSuite. You might still face scrutiny from Twilio if you send too many messages, but incoming should be no issue. Most of my clients do not use this application simply because the configuration is extensive. I would devote a couple of hours to it if you plan to test it. However, it works well once set up. More options are always a good thing. Full instructions, which change often, can be found on my site at [inteltechniques.com/VoIP.suite.html](http://inteltechniques.com/VoIP.suite.html).

## **Telephone Number Considerations**

Are you confused yet? With so many options, I find the complexity of choice within telephone communications to be a real issue. Twilio, MySudo, Google Voice, and traditional numbers present numerous usage options. Overall, I hope these previous guides help you determine your own usage strategy. However, I want to present one final summary of how I use these services for myself and clients. I think this may help your own decisions.

I carry a GrapheneOS Android mobile device while traveling. It has a SIM card with a true cellular number, but I never use it for calls or texts. I have Sipnetic installed on the device and four VoIP numbers configured. I can make calls from any of them, and all numbers ring directly to the device. I have an old Google Voice number which is required by some banks due to the history of use. I never make calls from it, but I have forwarded all incoming calls to a VoIP number. If my credit card company insists on calling me at a known number, I can receive a call through Sipnetic via VoIP, originally from Google Voice. Google has a log of the call, but no details about the conversation. All incoming text messages are forwarded to my email. I have MySudo on my devices which is used as previously outlined. I have over 30 total active numbers today including several old Google numbers.

I deviate a bit from my own strategy with clients. Many also receive a GrapheneOS device with Sipnetic, but most only need one Twilio number. They use it for all traditional incoming and outgoing calls and never use their newly-assigned cellular number. The number can be abused any way desired, and is the line used for all traditional phone calls. I then create a new Google Voice account. When prompted to enter a valid cellular number, I provide their previous true number associated with their old phone (which still has service). I then add the Twilio number to this Google account as a secondary number. When I am ready to shut off their old service, I port that previous cellular number to the Google Voice account. I configure this Google Voice account to forward any incoming text messages to an email account. This way, an incoming text to their old cell number is routed to email. This message can be read regardless of location.

The Google Voice number is provided any time a telephone number is required for two-factor authentication. The Google account is secured with a YubiKey. The likelihood of an attack toward Google is much less than the abilities with a standard cellular number. The text codes arrive securely within an email account, which can be accessed from anywhere. Google Voice also supports text messages from short code

numbers. Overall, try every service by taking advantage of free trials and identify the option best for you. Things change quickly with technology and you may find my results inaccurate.

### **VoIP Acceptance Issues**

VoIP numbers work great for incoming and outgoing calls. They can work well forwarding incoming text messages if you are willing to configure the options. Outgoing text messages can be a pain unless you are using MySudo. The real problems occur when an organization refuses to allow you to provide a VoIP number for services. Many banks require a true cellular telephone number in order to use their online banking. When you provide a VoIP number, you are likely denied the connection. If you try to provide a VoIP number during account creation with many social networks, you are declined an account. This is a constant battle, but I have some solutions.

If you ported your true cellular number to a VoIP provider, such as Google Voice, Twilio, or MySudo, that number will probably pass VoIP scrutiny for several months. This is because banks and other online services query the provider number through a carrier identification service. These are notoriously outdated and your ported number will appear to be associated with a true cellular provider for some time. Even though you may have ported a number from AT&T into Google Voice, the carrier ID will display AT&T until various databases are updated. I currently have a ported number which passes scrutiny on every online service I have tried (for now).

There is a lot to digest here. Take your time and determine the best path for your daily communications strategy.

# CHAPTER NINE

## DATA SERVICE

Data-only plans have been around a long time. These plans are typically sold by service resellers and are targeted toward vacationing travelers. If you live in the U.S. with a cellular plan through T-Mobile, you will need something else when you arrive in Grand Cayman for your 10-day stay. Likewise, residents of the United Kingdom will need something for their devices during their American holiday.

My interest in data-only plans stems from a privacy perspective. After witnessing numerous clients become the victim of SIM swapping in order to infiltrate their iCloud accounts or stalking by abusing cellular location data, I wanted to explore alternative avenues of cellular service. Since most clients never use their cellular-provided telephone numbers, and only need data for their secure communications, I took a close look at data-only plans.

There seem to be countless options, and many are simply not a great deal. My focus here will only involve services which I believe could be a potential replacement for a cellular plan. First, we should understand the person who would benefit most from this type of service. **Please read this entire chapter before executing anything!**

When you purchase a data-only plan, you are typically not issued a cellular number. A reseller has simply sold you the right to use a specific amount of pre-purchased data from a carrier or carriers. This is our first benefit. If my cellular plan possesses no telephone number, there is no way I could accidentally expose myself by using that number for communications or 2FA. There is no real threat of a SIM swapping attack since nothing would be gained. If you were to successfully steal my service, any other data plan would get me back in business. Without a number, there is no easy starting point for an attack against me.

The next benefit is the short-term nature of a data-only plan. I can easily purchase a plan with service for a day, week, month, or year. There is no real commitment. An even better feature is the multiple-carrier access. I can purchase data from a reseller who has connections with all of the major carriers. I can connect to AT&T, T-Mobile, or Verizon from the same plan. If one carrier has a weak signal at my current location, I can switch on the fly to the better option. None of the carriers know my true identity. They simply have a couple of unique identifiers from my device. While they log everything on their networks for years, I am a small needle in a very large haystack.

My search began with basic providers such as Tello and US Mobile, which were previously mentioned. Both offer data-only plans which do not provide a telephone number. However, these are single-service providers. You must pick the carrier you like and stick with it. I wanted more variety.

I then placed my attention on the international data plans. Most of these are operated out of Hong Kong, but provide service practically anywhere in the world. Those that

I tested worked fine, but the prepaid data usually expired every month. This led to wasted data and extra fees. More concerning was that many would not allow any type of private payment. Of those I tested, Keepgo was the most forgiving. However, their data was quite expensive.

All of the providers mentioned up to this point provided easy access. After purchase, I was issued a QR code which represented an eSIM. I scanned the code into my device and I had access. Nothing too special or burdening there. After playing with multiple options, I finally landed on a provider who checked all of the boxes for my privacy-seeking data-only service supplier: Twilio. However, that was short-lived.

Twilio recently announced that their data-only wireless plans were being acquired by Kore. The process to obtain an account with Kore is very invasive, and they demand to store a credit card and your true identity for any service plans. I do not recommend them. Today, our best option is Telnyx (<https://refer.telnyx.com/refer/zrfmo>). This affiliate link should provide \$20 in free wireless data access.

I mentioned Telnyx in the previous chapter. They are a DIY VoIP provider. I can purchase numbers and service easily at an affordable price. They also provide wireless services, often referred to as IoT service. They do not own any cellular towers or provide direct cellular connections. Instead, they have partnered with most carriers throughout the world. Their physical SIM cards are accepted by existing cellular networks, and service can be obtained almost anywhere in the world. You do not pre-purchase any data. You simply possess an account and pay for what you actually use. There are no expiration dates and no data is wasted. Does this seem too good to be true? Well, there are caveats of course.

The biggest difficulty is obtaining service. This was stressed in the previous chapter. If you devote the time to establish service as previously explained, you should not have any issues. Take the time to do it right. The reward is a telephony account which has many benefits. Once you have an upgraded Telnyx account with a balance, you have access to all of their services, including cellular providers.

Even with an active account, there are issues. The worst (and best) part of this plan is the cost. An active connection to almost every cellular provider in the world is as low as \$2.00 monthly. However, the data is not always cheap. On average, you will pay a range from \$0.04 to \$0.07 per MB in the U.S. That may sound inexpensive until your child decides to stream a 2-hour 4K video from your device. If that happens, you will regret reading about this strategy.

Let's work our way through the process and then I will provide some real-world examples. While researching this chapter, I ordered several Telnyx SIM cards. They arrived quickly and I began testing. I tried the first card within multiple devices, but it never worked. I contacted support but received no help. The same thing happened with a second card. I double-checked all settings, but never could make a connection. At times, my device informed me that the SIMs were "forbidden" on the towers of Verizon and AT&T. I contacted support again, but they could not help. They only sent me canned messages for 15 days.

Out of desperation and frustration, I reached out to the CEO directly. He insisted he would have this fixed. He had two engineers schedule a conference call with me. During the call, they asked me to try everything again, which did not work. They told me they could not resolve the issues with their own service and stated they would get back to me with a solution. Later, the only additional option was to try more cards and see if they worked. Finally, a third card created the desired connection. Today, I believe most Telnyx SIM cards will function as long as all proper settings are provided.

You can order SIM cards through the Telnyx portal. They are only \$1.00 each and include free shipping. Upon receipt, simply activate them within your Telnyx portal. Make sure you apply the following APN to the SIM settings within GrapheneOS.

Name: Telnyx  
APN: data00.telnyx

You should see a new shortcut within your GrapheneOS app drawer titled "Telnyx". This is not a third-party app. It is an embedded Android configuration for SIM connections. Make sure the "IMSI Selection Menu" is set to "Sparkle" and "Selection Mode" is set to "Manual". If these do not work, experiment with the other options.

Today, I rely on a Mint Mobile eSIM card for the majority of my connections. It is the default option within my device for calls, texts, and data. I travel a lot, and often find myself in remote areas with poor T-Mobile reception. When I do, I switch my data connection to the Telnyx physical SIM card by executing the following.

- Navigate to "Settings" > "Network & internet" > "SIMs".
- Select the "Telnyx" SIM.
- Enable "Mobile data" and "Roaming".
- Confirm my selections.

This then switches my data to Telnyx. My device searches for any AT&T, T-Mobile, US Cellular, or Verizon connection and allows me to use their data. For me, this is mostly about redundancy. I want to be able to always have a connection. I pay \$0.07 per MB of data used while connected through Telnyx for the first 100 MB. The price drops to \$0.04 per MB for the next 400 MB of data. It drops further after that. When I am back in a T-Mobile area, I re-enable my Mint eSIM through these same settings, and make sure my Telnyx SIM is completely disabled.

When I travel internationally, I experience the true power of this strategy. When I land, I enable Telnyx and immediately have service. It can be expensive service, but it does not require me to find a kiosk; purchase a new SIM; display my ID; and expose my identity. Always check pricing for the country you will be visiting before activating the service. Some tiers will charge several dollars per megabyte. My last trip to the U.K. was \$0.20 per MB.

I am very careful to only use this option when needed. My phone is in airplane mode most of the time. Only when I need to check my communications do I make a



connection, and I make sure to limit my activity. I do not browse the web, play videos, etc. While at the hotel, I take advantage of Wi-Fi from my laptop for all additional communications. My last 6-day international trip resulted in a \$9 data charge to my account. Best of all, I used the SIM connection already within my device.

I have clients who use this as their only source of data. They only have one SIM card within their device. They leave their devices in airplane mode until a connection is needed. After disabling airplane mode, they enable the SIM option and allow Telnix to connect. They then conduct their activity on the device and re-enable airplane mode. Some clients live in remote areas and only check their pending communication once or twice daily. They might spend \$0.40 per day. There is no cellular number assigned from any carrier; the carriers do not know the identity of the clients, and Telnix possess generic information within the account.

Is this the new perfect way to access cellular towers? I don't think so. It is one of many options. We are all unique and have our own needs. For me, it is an affordable backup connection and international solution. For most clients, it is a way to make sure they always have some type of coverage. For a few clients, it is the most anonymous way they can possess a connection to the rest of the world without jeopardizing their safety. I only ask you to consider all of your options and understand the risks and benefits of each. If you end up with a \$800 Telnix bill, don't blame me.

## **Warning**

If you are using Telnix as a data provider, or any other prepaid option, I highly recommend you disable all Google services which were discussed in Chapter Five. This prevents Google from downloading large amounts of data during data-only cellular usage. If you need push services while on these types of plans, I recommend you at least disable network access from Google Play. In order to make sure we are covering all of our bases, let's recap the previous steps and apply further restriction. I conduct the following on devices which will rely on prepaid data-only cellular access.

- Add "Data Saver" to the drop-down Quick Menu.
- Navigate to "Settings" > "System" > "System update".
- Choose "Permitted networks" and select "Not roaming".
- Open F-Droid and tap "Settings" and go to "Updates".
- Change "Over Wi-Fi" to the maximum level.
- Change "Over Data" to the minimum level.
- Disable "Automatically fetch updates".
- Close F-Droid and return to GrapheneOS.

If you have enabled push services, conduct the following. This will prevent Google from downloading updates to its various applications and services. This is not advised within most Android systems, but the frequent updates from GrapheneOS, which include any updates which have been pushed down by Google, make this acceptable in my view. Push services should still work without this access.



- Navigate to "Settings" > "Apps" > "See all..." > "Google Play Store".
- Select "Permissions" then "Network" and change to "Not allowed".

I only connect to these plans when I need to. I do not leave my device connected and absorbing data all day while it is in my pocket. My strategy, which occurs approximately ten times throughout the day while I am on these plans, especially during international travel, is as follows.

- Enable Data Saver mode.
- Disable Airplane mode.
- Conduct any necessary communications.
- Enable Airplane mode.

I then repeat this process whenever I am in a position to check my communications. Obviously, this prevents any notifications or incoming calls, but it also eliminates a lot of data waste.

Overall, if you do not need push services, and you have controlled the way GrapheneOS and your application managers apply auto updates, and you will allow updates over Wi-Fi, your device should use very little data during normal activity. If you have push services enabled but you disabled network access to Google Play, you should only see a slight increase. When I rely on data-only prepaid service, I make sure Google is completely disabled. Consider the following.

While testing Telnix cellular service, I enabled all Google options within GrapheneOS. While monitoring data access by my email and messenger apps, I witnessed a 60 MB download via Google Play. It decided to update its software without warning or consent. The same update would have been applied by GrapheneOS during my typical weekly update over Wi-Fi if I had waited. This is why I insist on disabling network access from Google Play store on any client devices which possess Google services and connect to data-only providers. It is also one reason I completely eliminate Google from my personal device, especially while traveling overseas.

## Summary

I believe all international travelers should embrace data-only SIM connections as a secondary account on their devices. Non-travelers who never leave the country might find them valuable as secondary connection options when poor coverage is an issue. On some occasions, high-target individuals may rely on them as their primary cellular access. Understand all options and choose the path best appropriate for you. Most importantly, know the costs of the data and the ways your device might download unnecessary data.

While writing this book, I have been creating a test device with all services mentioned. At this point, I have the following configuration.

- I have a brand-new device with a secure and private operating system.

- I have a traditional cellular plan through Mint (\$15) connected via eSIM.
- I leave it active as my primary service for calls, texts, and data.
- When I am in the U.S. and have poor T-Mobile coverage, I can switch to the physical Telnyx SIM card and access data from any carrier at \$0.07 per MB.
- When I am outside the U.S., I can activate my Telnyx SIM and access data at \$0.20 or more per MB.
- I have an eSIM Tello account with voice and text service which I can use when a company demands a true cellular number (\$5).
- I have Wi-Fi calling enabled through Tello which allows me to use this number while in airplane mode.
- I have both Twilio (\$1.15) and Telnyx (\$1.00) VoIP numbers programmed and available through Sipnetic.
- SMS messages to each forward to my email.
- I have a MySudo Pro plan (\$5) which gives me three numbers with full access to calls and SMS.
- I have two Google Voice numbers (\$0.00) forwarded to my VoIP options.

You can now call me through nine different numbers and my phone will ring. I have four unique service providers available at all times. This may sound costly. The scenario I have described here has a monthly charge of \$31.00 plus minimal VoIP usage. While writing this I asked a random family member what they pay for their one service which provides one number. She responded "About \$45".

My personal device possesses over 25 active numbers with four cellular providers, and I never pay more than \$40 monthly for everything. That is overkill for most readers, but I am a telephony nerd. I rely on Mint for daily data usage, Telnyx within poor T-Mobile coverage while in the U.S. and while outside of the country. I rely on VoIP numbers for all unencrypted calls. For everything else, I rely on secure communications, as explained next.

# CHAPTER TEN

## SECURE COMMUNICATIONS

You should now have a new device that has no connection to you. It possesses prepaid cellular service with no name attached. Since you do not use the cellular number provided for any communications, the carrier has no log of your calls and messages. If I wanted to attack you through your mobile device, I have no information to begin my hunt. All of your outgoing calls are made through VoIP numbers, which may not possess your true identity. While any mobile telephone is a tracking device which always possesses some type of digital trail to the owner, you have created numerous layers of privacy which will keep you protected from traditional attacks and monitoring. We now need to harden your communications.

Over 95% of my daily communications occur over secure channels. I use software and services which provide End-to-End Encryption (E2EE) to protect my conversations and the storage of data. This applies to email, voice, video, and instant messaging services. For email, I rely on Proton Mail and Tutanota. If I send you an email from my Proton Mail account to your Proton Mail account, it never leaves their network. Even if forced by a court order, the encryption prevents any employee from seeing the content. The same applies to messages sent from one Tutanota account to another. The key to the security is not leaving the network. An email from my Proton Mail account to your Gmail address has lost all security. There are many considerations here, including the following.

- **Secure Messaging:** There is nothing I can say about secure messaging applications that has not been said elsewhere, and I suspect that anyone interested in privacy has already adopted a favorite service. However, a book on mobile devices would not be complete without mention here. Standard SMS text messaging leaves metadata within the systems of your cellular provider, and they can access the content of the messages. Cellular companies store years of this data, which can then be released intentionally or accidentally.
- **Zero knowledge, End-to-End Encrypted (E2EE):** This means that all communication is completely encrypted and even the provider cannot allow the content to be intercepted in any way. Trusted providers have no ability to view the contents of your communications because the level of encryption from your devices prevents them from any ability to access your data.
- **Ephemeral Message Expiration:** SMS messages leave a history with cellular companies. Secure communication services give you more control. Reputable services allow you to set an expiration of your messages. Once the expiration passes, the messages disappear on your device and the recipient's device. This is not bulletproof, as screen captures or exports can create additional copies, but it provides a basic layer of protection.
- **Encrypted Voice Calling:** When I need to talk with a client, I only use services which provide true encrypted calling. This prevents network wiretapping and other technologies from intercepting and recording my call. There is still a risk

that the other party could record the conversation, but interception by a third-party is unlikely. A telephone provider can intercept any call.

- **Adoption:** If no one else in your social circle is using your favorite secure communications application, then it is useless. The security only works for communication within the network. Services with a high adoption rate will always be preferred over niche applications with minimal users. There are many secure messaging apps emerging every day. I will disclose those which I use and recommend and those which I believe should be avoided.

## **Secure Communication with Signal**

There are things I do not like about Signal ([signal.org](https://signal.org)), but it has the largest user base and is therefore my primary secure communications platform. There is a decent chance that many of the people in your circle already use the service. I would rather communicate over Signal than SMS text, and most people in my life possess Signal as their only secure option. I have great faith in their encryption protocols used to protect my communications from any outside party. Unfortunately, Signal prioritizes mass adoption and unnecessary features over extreme privacy, but we will make it work well for our needs. Let's tackle the biggest issue first.

Signal requires a telephone number in order to create an account, which is a huge privacy violation. You must then give out this number in order to communicate with others securely. This shares your number in a way we typically try to avoid. If you choose to use Signal, you should create an account associated with a VoIP number, as previously explained, such as a Twilio, MySudo, or Google Voice. I typically prefer to use a client's previous personal number which has been ported to Google Voice for this use since it may already be known by others in their circle. This shares the VoIP number with all contacts, but that does not expose the new true cellular number. Using this old number can make communications easier and more trusted by the other party. Never use your true cellular number with Signal.

Signal notifies people when one of their contacts creates an account. This may be beneficial to you if your ported Google Voice number is already trusted by your friends. If you do not like this feature (I do not), you might consider using a brand new VoIP number unknown to anyone else. This eliminates any contacts knowing you are now on Signal. I created a VoIP number which is only used to establish communications with others through Signal. This may be unnecessary for you. Let's walk through a typical configuration of Signal.

- Download the Signal app through Aurora Store.
- Launch the app and accept the default requirements.
- Enter a VoIP number and confirm a text message or voice call.
- Provide a desired first name, which can be a single letter.
- If prompted, enter a secure PIN.

Some users will need to tap the alert about missing Google services. Select "Allow" if you want the app to always run in the background and receive notifications of

messages. Tap "Deny" if you want to preserve minimal battery life and retrieve messages only when you open the app without notifications. Those who enabled push services will not have this issue.

Once you have an account, you have access to secure (encrypted) text, audio, and video communications, including group conversations. Signal has a desktop application which supports all features available to the mobile version, which we will install in the next chapter. If you are using GrapheneOS, Signal may be the only messaging application which will reliably send notifications of received messages. If you have children or other family members which need immediate access to you, then I highly recommend configuring Signal on their devices. This will ensure that you do not miss important messages due to the potential lack of Google services on your own device. It will also introduce secure communications to the family. Let's configure a few more settings to make things more private.

- Open the "Settings" menu by tapping the icon in the upper left of Signal.
- Tap "Account" and enable "Registration Lock". This requires your Signal PIN to register a new device.
- Tap the back button and open the "Chats" menu.
- Disable everything in this screen.
- Tap the back button and open the "Privacy" menu.
- Disable "Read receipts" and "Typing indicators" if desired.
- Set a desired time for messages to disappear.
- Enable "Screen lock" if desired, which forces a fingerprint or PIN to open.
- Click "Advanced" and disable "Show Status Icon" and "Allow from Anyone" if desired.

Signal is far from perfect. Many elitists insist on using robust apps such as Session and avoid widely-adopted services such as Signal. I understand the desire for extreme privacy, but we must always place emphasis on products which our contacts will actually use. My entire family made the switch to Signal because it was quite easy for them. They did not need to memorize an additional username and password. They simply connected the account to their true cellular number which they have had for many years.

Privacy and security are likely not as important for everyone in your life as to you. We must choose our battles wisely. If your non-technical contacts are willing to use Signal but do not want to fuss with more complicated options, I still consider this a win. Your conversations are encrypted and much more secure than any traditional protocol, such as SMS.

### **Secure Communication with Molly**

Some readers may have a use for a second Signal account. The official app allows only one instance of Signal per device, but we can use a fork of Signal called Molly if another account is needed. If interested, conduct the following.

- Navigate to molly.im in your browser from your device.
- Tap "Molly F-Droid Repo".
- Choose from "Molly" or "Molly FOSS". The FOSS version uses its own push service while the non-FOSS allows Google's push services to be used.
- Tap "Add" to add to F-Droid.
- Open F-Droid.
- Search Molly and install the application.

You can now configure a second Signal account in the same way as the previous tutorial. I use Molly for communications with clients and reserve Signal for family and friends.

### **Secure Communication with Wire**

Wire (app.wire.com) is my second preferred secure messenger over all others. While not perfect, it offers features currently unavailable in other providers. Wire is free for personal use, and has adopted a large audience of users within the privacy community, but it is usually ignored by the masses which flock to Signal. Only an email address is required to create an account, and I recommend Proton Mail for this purpose.

GrapheneOS users can download Wire through Aurora Store. You can communicate securely via text, audio, and video across all platforms. This is a rarity and makes the service easily accessible in any scenario. I often provide existing Wire account details to a new client, which allows them to open a browser and immediately connect to me without creating their own account. This has been very valuable in my line of work.

I do have minor complaints about Wire. First, I have witnessed messages appear within the mobile application but not the desktop or web versions. If I search for the user, I then see the text content, but this can be a hassle. This only applies when the desktop or web versions are closed. When they are open and active, the messages appear fine. Fortunately, deleting a message on one device removes it from all. Signal does not offer this.

Installation and configuration of Wire is much more straight-forward than Signal. Download the app; create a "personal" account; and share your chosen username with others. Click the silhouette icon in the lower left to search for a user and initiate a text, voice, or video conversation. One unique feature of Wire is the ability to configure up to three user accounts within the desktop application (two on Android). On both my mobile and desktop versions of Wire, I have the same accounts which I can use for various purposes. This alone justifies Wire as one of my preferred services. If you have push services enabled, Wire notifications will arrive as normal.

Some may question my endorsement of Wire. In 2020, they transitioned their company headquarters from Switzerland to America. This immediately triggered those who distrust 5-eyes governments. In this scenario, you would also not want to use Signal, MySudo, or most other secure messaging options. I am not concerned with the location of their headquarters. I am more interested in the security of their product

and encryption protocols, both of which I trust. Both Signal and Wire have completed numerous third-party security audits, all of which are publicly viewable online. These audits will always outweigh the location of a team or building when I consider use of a secure product.

### **Secure Communications Summary**

Overall, you should adopt whichever secure service will be used by those in your circles. If no one in your life is using secure communications, you have an opportunity to select the best service for your needs and start recruiting people to it. If everyone in your life already uses a specific service, jump on board. I have great respect for many other secure messaging applications, but various reasons have prevented them from appearing within my primary recommendations. Consider the following.

- MySudo (mysudo.com) offers free secure communications within their network. This includes E2EE text, audio, and video. If the majority of your contacts already have MySudo for their VoIP solution as previously explained, then this may be the only secure option you need. It did not make the "top two" because of lower adoption and no ability to place calls or messages through a browser or desktop application. This is vital for clients who do not bring a mobile device into their homes.
- Session (getsession.org) has very private text messaging options, but adoption is extremely low and voice calling is not supported.
- Matrix (matrix.org) is a phenomenal open-source and decentralized platform, but their focus is on community chat rooms for a niche tech-savvy audience.
- Threema (threema.ch) meets all of my requirements with exception of adoption. Their paid app is justified, but payment prevents many people from downloading it.
- Jitsi (jitsi.org) possesses a great video conferencing protocol, but few people use it for traditional text communication. I use this weekly in place of Zoom, but never for text.
- NOT RECOMMENDED - Wickr was the first secure communications app I ever used. However, I stopped using it in 2020 when I discovered that they were sharing user details with third party services including Microsoft and Google. The CTO of the company confirmed analytical data and IP addresses of all users are shared. In 2021, they were acquired by Amazon. I have deleted the app.
- NOT RECOMMENDED - WhatsApp provides secure end-to-end encrypted text and voice communication with a very trusted protocol. However, the service is owned and operated by Facebook. Furthermore, a privacy policy shift in 2021 allows them to share account details with Facebook servers and users. While the company says this is isolated to business Facebook profiles who wish to incorporate secure communications with customers, I have no room for this product in my arsenal. Furthermore, their user backups are not encrypted and often stored within Google cloud products.



- **NOT RECOMMENDED** - Telegram supports E2EE communications, but the setting is optional. The default configuration potentially exposes content internally. I never rely on a communication platform which requires user customization to make the content secure.

### **Secure Communications Conversion**

You may have found your desired secure communications platform. Now what? If no one within your circle of friends and family uses it, it is of little use. Obviously, a polite request and explanation of the benefits may draw a few people in, but you may want to convert all of your contacts over to something secure. That is what I did. I first asked each person within my immediate communication circles if they had a preference of a secure communications provider. If they did, I connected with them through that option. If they did not, I asked them to download Signal. This is because Signal is the easiest to implement without any requirements for a username, email address, or password. It just works. Most people do not have any issue using their true cellular number for this purpose. Although Signal is easy to install and execute, some people simply will not transition to it in order to communicate with you. When you encounter these scenarios, consider the following conversion strategies.

**Response Delay:** When a contact refuses to adopt a secure messenger, and only sends messages via SMS to one of my VoIP numbers, I politely explain the ways in which SMS is insecure. If that does not help, I place them on a delay. When I receive a message via SMS, I do not respond for at least 24 hours. I then state "Sorry, I rarely check SMS, contact me on Signal if you need anything immediately". If they contact me via Signal, I respond right away to reward the attempt. After a few weeks, they only contact me on Signal.

**Missed Connection:** I once had a close friend who simply refused to use anything secure. He had downloaded Signal but never opened it. He would send sensitive communications over SMS which I found troubling. The delay option did not work on him. Therefore, I had to get creative. This friend was a huge 80's rock fan. On the night which his favorite band Def Leppard was in town, I sent a message via Signal asking him if he wanted free front row tickets to the show. I knew he would not check and respond, so I was not too worried about my bluff. A week later, he noticed the pending message notification for Signal and read the message. He was very regretful, and began checking Signal more often. The next time I had a true offer for free entertainment, I reached out via Signal and we met up.

**Daily Reward:** The most difficult conversion has been my extended family. My siblings joined right away, but some family members were hesitant. I was able to convince them to install Signal, which allowed me to add them to a group conversation, but they did not open the app often. My solution was to create a new group of all immediate relatives, and engage them in a daily chat. I identified the people who were not seeing the conversation, and started sharing old family photos of them. Childhood pictures of myself and other relatives at holidays generated a lot of conversation around the memories of our past. Those who were participating then copied some of the images

to others who were ignoring Signal, which immediately encouraged them to launch the app and see what else they were missing. I have found that sharing old family photos is a great way to draw people in. If you are uncomfortable sharing images of yourself or others within secure chat, consider ancestral images. Every week, I post random photos of my deceased grandparents to my sisters in a secure group chat. This not only presents an opportunity to bond over memories, but it also creates a pattern of behavior which encourages daily use of the app. If desired, you could set a timer for the images which makes them disappear after a set amount of time. This encourages people to look right away.

I warn readers to avoid secure messenger switching fatigue. I am guilty of this. Many years ago, I asked people to join the secure messenger called Wickr. Once I realized they were collecting user analytics and forwarding to Microsoft and Google, I asked them all to switch to Wire. Once I began using Signal heavily, I asked the same people to switch to that. This creates an annoyance and discourages people from playing along with your antics. Choose the most appropriate option first and test everything. Make sure you are comfortable with the product and are confident in its long-term availability. Only then, invite people in, and do not ask them to switch unless there is a good reason.

# CHAPTER ELEVEN

## VPN CONFIGURATION

Virtual Private Networks (VPNs) provide a good mix of both security and privacy by routing your internet traffic through a secure tunnel. The secure tunnel goes to the VPN's server and encrypts all the data between your device and that server. This ensures that anyone monitoring your traffic before it reaches the distant server will not find usable, unencrypted data. Privacy is also afforded through the use of a distant server. Because your traffic appears to be originating from the VPN's server, websites will have a more difficult time tracking you, aggregating data on you, and pinpointing your location.

Virtual Private Networks are not a perfect anonymity solution. It is important to note that VPNs offer you privacy, not anonymity. The best VPNs for privacy purposes are paid subscriptions with reputable providers. There are several excellent paid VPN providers out there and I strongly recommend them over free providers. Free providers often monetize through very questionable means, such as data aggregation. Paid VPN providers monetize directly by selling you a service, and reputable providers do not collect or monetize your data. Paid providers also offer a number of options which will increase your overall privacy and security.

I currently use and recommend Proton VPN as my primary VPN and Private Internet Access (PIA) as a limited secondary dedicated IP option when VPNs are actively being blocked. I explain this more in a moment, but I always keep details and discounted purchase links at <https://inteltechniques.com/vpn.html>. Purchases include unlimited use, connection to multiple devices simultaneously, and fast speeds. The following are my affiliate links which support the free podcast.

Proton VPN: [https://go.getproton.me/aff\\_c?offer\\_id=26&aff\\_id=1519](https://go.getproton.me/aff_c?offer_id=26&aff_id=1519)

PIA: <https://www.privateinternetaccess.com/ThePSOSHOW>

Many privacy purists recommend Mullvad as the best VPN provider, but I have consistently had bad experiences. The speeds were slow and connections were unstable. Therefore, I do not use or recommend them.

My VPN policy is quite simple, but my opinions about VPN companies can be complex. Any time that I am connected to the internet from my laptop or desktop computer, I am connected through a VPN. I know that my internet traffic is encrypted and originating from an IP address not associated with me. I never want websites I visit to know my home IP address, which would be disclosed in their next breach. I protect my entire home with a pfSense firewall, which is explained within *Extreme Privacy, 4th Edition*.

If I am using my mobile device from home on Wi-Fi, it is automatically protected by my firewall VPN, and a secondary VPN app is not needed on the device. This is where my opinions on VPNs may not be well received. Most of the time when I am traveling

with my mobile device connected to a cellular network, I am not connected to a VPN. Before the extreme privacy readers become upset, please allow me to explain myself.

When I am away from my home, my device is on and connected to my cellular provider. My location at all times is known because of this connection. A VPN would not help with that. My service is in an alias name and I adopt more of a "pull" instead of "push" digital lifestyle, so my device is not extremely active in the background.

When I stop to check my email, my provider knows that Proton Mail is being used on the device. I do not see much harm in that. The data is encrypted and my carrier cannot digest any content. They only know the domain or IP address to which I am connecting.

Proton Mail knows the true IP address of my device. It is likely an IPv6 address which is also being used by other people on that network. It is registered to my cellular provider, and does not identify me, my account, or my home. However, my general location on a city level might be known. Again, this is not a huge threat to me.

I do not browse the internet, play video games, or communicate through my carrier-provided telephone number. I do not object to my carrier knowing my device uses Proton Mail, Signal, or other secure messaging applications. I know my data is safe. I do not have any Google or Apple services sucking up my IP address and activity for their benefit. My device is fairly basic.

This is not appropriate for everyone. I have clients who leave their mobile device VPN activated at all times. They are under an immediate physical threat and cannot allow their adversary to know anything about their true location. They would never allow any secure communications providers to know which state they were in, and do not allow their provider to know which services they use. If you have a tech-savvy stalker, a constant VPN might be a requirement.

My point is that we are all different. We all have unique scenarios which demands a unique response. You must identify what is important to you. Do you barely use your device and switch providers often like I do? Then a VPN might not be at the top of your list. Do you have someone in your life who seems to always know where you moved and your new number? Then a VPN might be a high priority. Whatever your scenario, we should understand some VPN usage strategies.

I possess two VPN applications on my mobile device at all times. Proton VPN is my primary provider, and I use it when I want to hide my activity from my cellular provider or mask my IP address from a specific site or service. I can choose a server location which matches my desired presence and the connection is stable. I do not mess with the connection settings and allow the application to control all protocols. Since I use DNS-based filtering, I do not have any conflicts with various firewall applications.

The second app I possess is PIA. This always upsets some readers. PIA is a huge VPN conglomerate now and the executives have a colorful past. None of that concerns me because I only use this service for one specific purpose. Sometimes, I need a dedicated

IP VPN, which is explained in a moment. Since I purchased my PIA account with Bitcoin in an alias name, and their service has been audited by third parties, I have little concern about their privacy promises. There are very few VPN providers who offer a dedicated IP option, and I believe the way PIA anonymously provides this static address is done better than the others. If Proton VPN offered a dedicated IP address, I might ditch PIA altogether. Regardless of your VPN provider, let's talk about usage.

Many VoIP providers have issues with VPNs, yet some require them. If you have an always-on VPN within your mobile device, all of your traffic is being routed through additional servers. This can cause connection stability issues when placing telephone calls over VoIP. I witness more dropped VoIP calls while connected to a mobile VPN than connections directly to the cellular provider. When I need to make a call from my device, I typically make sure my VPN is disabled.

There are other times a VPN is required to make calls. Services such as MySudo only function within the U.S., Canada, and U.K. due to strict telephony regulation in other countries. When I am outside of these locations and need to make a call, it fails. This is because MySudo must block the IP addresses of many countries in order to comply with laws. If I am in Italy and need to make a call, MySudo fails me. However, I can launch the Proton VPN app, connect to a server in New York, and then complete the call. Since my IP address appears to be within the U.S., the MySudo servers do not block the call.

I have also witnessed some VoIP providers block my calls when I am within the U.S. when the IP address assigned to my device has been mistaken for a foreign address even though it was local. Having a VPN ready to go can get me past that hurdle. Overall, I never use a VPN while making VoIP calls within the U.S. unless forced. I always use a VPN while making VoIP calls while outside the U.S.

One major issue with VPNs is that many online services block them. If you are connected to a VPN from your mobile device, services can detect this and block access. I see this most often with financial apps. This is where a dedicated IP address can be beneficial. Consider the following scenario.

I previously mentioned my client who must be able to deposit checks for her business from her mobile device. She needs a true cellular account in order to receive a verification text upon each mobile login. She also wants to mask her location and IP address from her bank. Sometimes, she needs to deposit checks while she is at home and would never share her home IP address with the financial institution. If she connects from her home network, which is behind a firewall VPN, she is blocked. This is because her bank does not allow customers to access their accounts while behind a known VPN.

She launches the PIA app and connects to her dedicated IP address. She is the only person in the world allowed to use this address, which does not appear on the publicly-available VPN block lists. Once she is behind her dedicated IP, she can open the banking app; receive her true cellular 2FA while Wi-Fi calling is enabled; and complete her transaction. This is a lot of unnecessary work, but it is the world we live in.

This dedicated IP address can have many other uses. When I am on my desktop computer and need to log into my bank, I have to activate my dedicated IP or I will be blocked. Any site which blocks VPNs can usually be bypassed with this feature. If you adopt the dedicated IP address option, make sure that you change the protocol to OpenVPN; Transport to TCP; and Remote Port to 443. This will help satisfy the banking apps.

For most readers, and almost every client I have consulted, I recommend sticking with the standard application provided by the VPN company. These branded apps should suffice for most needs. Both Proton VPN and PIA can be downloaded from Aurora Store. Once installed, simply provide your account credentials and launch your VPN connection. Fortunately, Proton VPN has made their applications completely open-source. This makes it much more difficult to hide malicious programming within them.

Please note that if you are using a VPN application on your device, it will likely ignore any DNS modifications made and use its own server. This is acceptable for many situations, but not ideal for everyone. I disable DNS in both of my VPN apps and rely on NextDNS as previously explained. By allowing your VPN to secure all traffic and provide all DNS queries, you are placing all of your eggs within one basket. You are trusting your VPN provider with the ability to log all of your internet history and traffic. This is probably not a huge threat if you are using a trustworthy VPN, but we can do better.

What do I use? I rely on Proton VPN through their app on my mobile device and laptop while I am traveling. Home devices are protected through a firewall with Proton VPN, as explained in *Extreme Privacy, 4th Edition*. I trust them more than most commercial options and I believe their business model is the most transparent. Being hosted in Switzerland provides some aspect of privacy from vague government intrusion, but international servers could always be compromised. Any updates in regard to my VPN recommendations will always be posted on my website at <https://inteltechniques.com/vpn.html>.

# CHAPTER TWELVE

## DEVICE CUSTOMIZATION

Your mobile device will probably be slightly different than any other device in the world. We all have preferences and customizations which make the device unique. These modifications could be purely cosmetic or there may be functions which we find beneficial. This chapter presents some ideas which might make your device more useful for your daily usage. Let's start with cosmetics.

If you are happy with your home screen and application drawer after configuring your GrapheneOS device, there is no action to take. I am very picky about the overall look of my device, so I make some drastic changes. I adopt a custom launcher to replace the default GrapheneOS Home app. I do this so I can change the look of the icons, modify the names of the shortcuts, and fit more information within my screen.

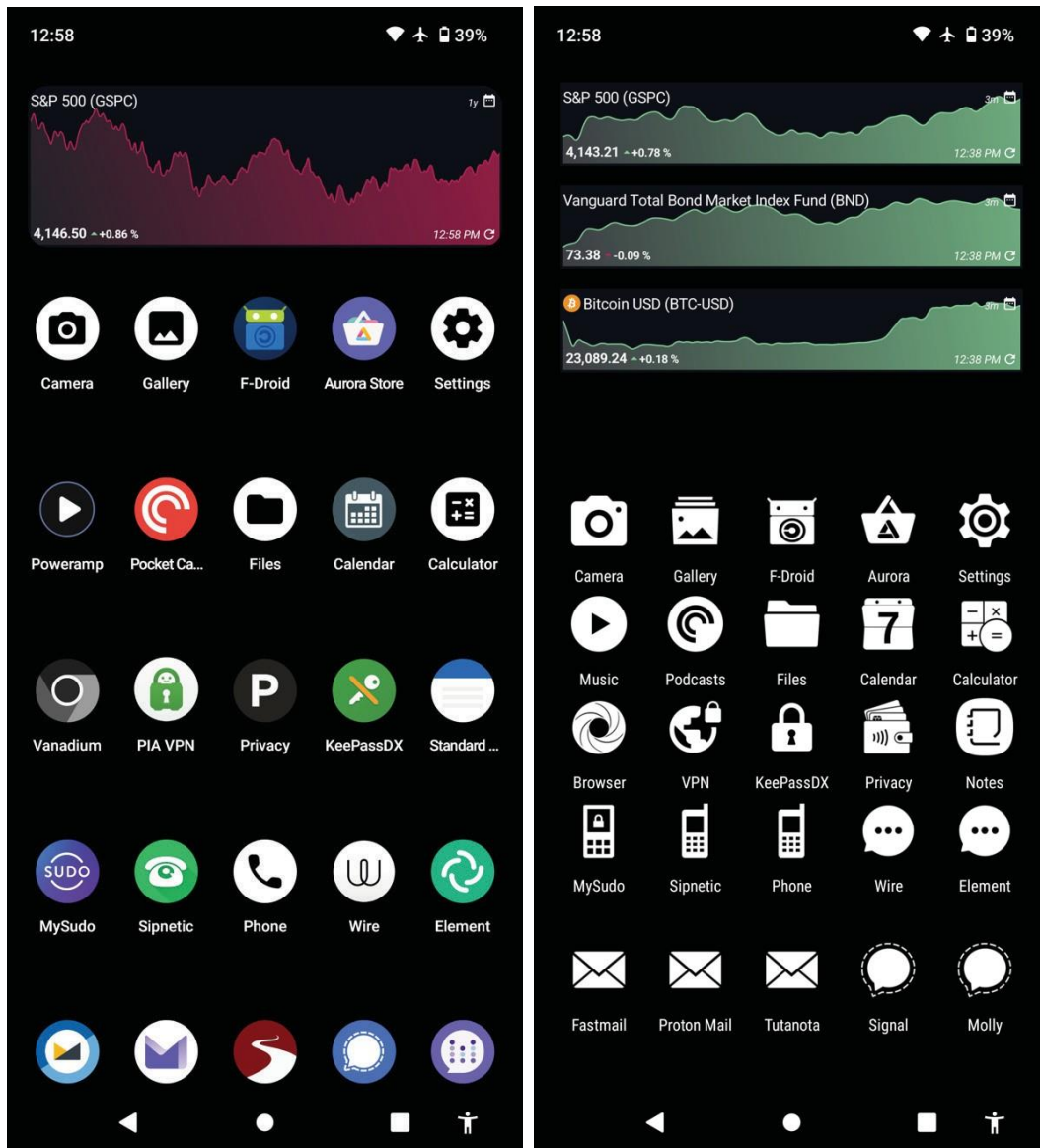
There are many custom launchers to choose from. Maybe too many. Nova Launcher is a staple within the customization community, but I do not use it. It is more robust than what I need and some features I want require a paid license. For the past few years, I have preferred Lawnchair. However, that is another complicated topic. Searching Lawnchair in F-Droid presents the original abandoned application which does not include major revisions received at the end of 2019. Searching it in Aurora Store presents Lawnchair 2 which is also no longer maintained. Searching within a browser presents Lawnchair 12 which is the active project, but it is still in Alpha testing. Which should you choose?

If you decide to use Lawnchair as a custom launcher, I currently recommend Lawnchair 2 available via Aurora Store. The F-Droid version is very outdated and the latest community version is not stable. Once Lawnchair 12+ is out of Alpha and Beta stages, I would consider switching to that, but I believe that is a long way out.

I downloaded Lawnchair 2 to my GrapheneOS device and began the customizations. After installation, I navigated to "Settings" > "Apps" > "Default apps" > "Home app". I selected Lawnchair and returned to the home screen. It appeared quite different. I swiped up to see the application drawer and selected Lawnchair to browse through the configuration options. The major settings I modified were the desktop icon grid (8x5), notification count (enabled), dock labels (enabled), and search bars (disabled). This provided me a good starting point.

From there I installed the Whicons icon pack from Aurora Store and activated it within Lawnchair. This allowed me to select new icons for all of my apps which are cohesive to my desired look. I could then tap and hold any app icon and choose "Customize" or the edit icon. From there I could change the icon and modify the app name. Instead of identifying my preferences, consider the following images.





The image on the left is a stock GrapheneOS home screen. The app spacing is similar to stock Pixel; the dock labels are hidden; and the app labels are often truncated. This all bothers me. The image on the right contains more icons with less spacing; the dock labels are visible; the app labels are modified; and the icons are all similar from an icon pack. The area for widgets on the left image is minimal, and I could only fit one. The image to the right displays resizing and placement of additional options.

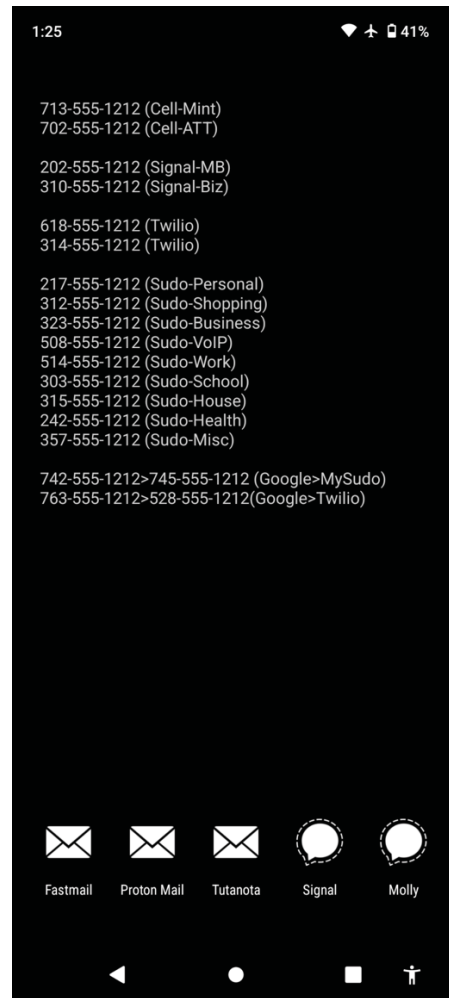
There are countless modifications you can make using a custom launcher. I only present this section to motivate you to identify the best launcher application and settings which makes your experience better. You might notice that Lawnchair 2 has not received any updates in over two years. This is because the team split and abandoned the original project. Since I disable network access when I install it, I do not have much concern about this. If this bothers you, you should seek an active project. I like the simplicity and stability of Lawnchair 2 over the more complicated modern alternatives. You might feel different. Many people use Nova Launcher.

You might notice some interesting choices within my examples on the previous page. Please keep in mind that these were taken during testing and do not reflect my personal daily device. This presents a good time to talk about application choices which have not been discussed yet. The following offers considerations for your own device.

- Password Manager: If you have adopted my offline password manager lifestyle with KeePassXC, I recommend KeePassDX as a mobile option. It works well with your existing KeePass databases. If you prefer a password manager with synchronization capabilities, I recommend Bitwarden.
- Notes: I rely heavily on Standard Notes for everything from encrypted notes to 2FA tokens. I synchronize my account to all of my devices. However, I also download Simple Notes for widget access, which is explained in a moment.
- Music: If you only need to play a few local MP3 files without any audio customization, then any free open-source player will work fine for you, such as Simple Music Player through F-Droid. However, I prefer Poweramp. It presents many customizations and a great database to present your media library. The EQ is one of the best I have seen if that is important to you.
- Podcasts: In the images above, you can see I was testing Pocket Casts. However, the telemetry and push for an account turned me off. I currently use and recommend AntennaPod. It is simple and just works. It can be downloaded via F-Droid.
- Stock Widget: I doubt many people care about hourly updates to the stock market, bond market, or Bitcoin price. If this is of interest to you, I recommend Stock Widget from Aurora Store. You can see it in action within my previous images.

If you have followed my tutorials and created numerous cellular, Signal, and VoIP telephone numbers, it may be hard to keep them all straight. There have been countless times when I have been somewhere and asked for a contact telephone number, only to fumble on my device to start looking at my options. Because of this, I place a cheat-sheet on my device and the devices of every client. I first download and install Simple Notes Pro through F-Droid. I then tap and hold the home screen in order to see the widgets menu. I then add a notes widget to the second screen of the home screen. This is accessed by swiping to the left in order to navigate to the screen to the right.

The following image displays a fictitious example. The first two lines are the cellular numbers for my device, which I avoid when possible. The second two options display my two Signal accounts. The third group contains two Twilio numbers ready for calls through Sipnetic. The next group is my nine MySudo accounts and the general purpose of each. Finally, the last section displays two Google Voice numbers which are being forwarded to VoIP options. If I were to display my true device, it would contain a lot more numbers. More importantly, one could call any number listed on my home screen and my device would ring. Having this list available allows you to quickly look at your home screen to pick a number without appearing suspicious. I use it almost every day.



## Profiles

GrapheneOS supports multiple profiles within a single device. This allows you to create unique configurations for multiple users, or your own alias profiles. I played with this for a few weeks, and found it very intriguing, but ultimately decided not to use this feature as part of my communications strategy. Since GrapheneOS is not "calling home" and sending our data out to Google or Apple, I found little reason to isolate my app usage. The one benefit I enjoyed was the ability to possess additional instances of Signal within a single device, but switching profiles to take advantage of this became tiresome. Overall, it is not for me but may be valuable to you.

Every modern Android device possesses the ability to create multiple user profiles within a single device. This allows you to create numerous environments which isolate apps and services from the primary profile. These are not virtual machines or completely restricted containers, but they do offer some privacy and security benefits. I believe they are best explained with a recent example of usage.

A client who possessed a private and secure device returned with a new problem. She absolutely needed Google Maps for daily navigation. However, she did not want any

Google services within her primary device profile. I created a second profile for this purpose by conducting the following.

- Navigate to "Settings" > "System" > "Multiple Users".
- Enable the "Multiple Users" toggle.
- Click the "+" to create a new profile and title it "Travel".
- Allow the device to reboot into the new profile.

This booted me into a new copy of GrapheneOS. All of the standard apps and services which were included with the device upon first boot were present exactly as they appeared on the first day. I followed the previous tutorials to activate the Google framework and install Google Maps via Aurora Store. The application worked flawlessly and turn-by-turn navigation was precise. She could share her location to this single application within an isolated profile without jeopardizing her other work. However, there are always caveats to this.

Exiting a secondary profile does not shut down all of the active services within it. You may notice an unnecessary burden to your device's RAM and battery if you leave this profile running in the background. The solution is to completely reboot after using the secondary profile or use GrapheneOS's "End Session" feature. My preference is to enter the new profile from your primary profile; conduct the required business; and then reboot the device. If you do not enter the secondary profile after a reboot, those resources should not be loaded. "End Session" also accomplishes the same thing.

This secondary profile is not anonymous or completely disconnected from the primary user. They both share the same stored Wi-Fi connections, cellular device, GPS, Bluetooth, and hardware identifiers. Our purpose of a secondary profile is to provide an invasive Google environment when needed without compromising your primary profile. A Google account is not required.

I never recommend more than one secondary profile. Possessing multiple profiles requires additional storage and maintenance. Forgetting to reboot after access of each profile could quickly drain your resources and drag down your speed.

## **2FA**

Every mobile device should have the ability to act as second-factor authentication (2FA) when logging into a service. In years past, this involved a SMS text message to your cellular telephone number which contained a one-time use code. Today, we can do better. My preference for 2FA options is generally hardware tokens first, then software tokens, and finally SMS text messages. On desktop systems, I rely heavily on a USB YubiKey device, but I rarely use one on my mobile device. Therefore, we should have redundant alternative options ready at all times.

First, we need a software-based 2FA token application which contains all of our seed codes which were supplied by various services when you enabled 2FA. These are the unique codes which allow the application to generate 6-digit codes which change every

thirty seconds. If you already have a service you prefer, simply install that application within your new device and synchronize your account.

If you are starting over, I would consider a paid version of Standard Notes. I already rely heavily on this encrypted note-taking application for notes, spreadsheets, and tasks. Adding their 2FA option makes sense to me. This synchronizes my 2FA codes across all devices for cross-platform support. Since Standard Notes recently adopted hardware 2FA via YubiKey, you can secure your account without the need for a secondary software token application.

If you do not want to pay for this service, the free version of the password manager Bitwarden provides the same feature. Since this is not a book about password management and 2FA, I will not provide full tutorials on usage. Simply know that you have multiple options for storing and displaying your own codes.

Some sites and services refuse to allow secure hardware and software tokens for their 2FA. While frustrating, this is a reality. Some sites, especially financial institutions, force us to use traditional telephone SMS text messages to receive our confirmation codes. When this happens, I attempt to use a Google Voice number first. That may seem inappropriate for a book about extreme privacy, but I believe Google Voice accounts are much more secure than any traditional VoIP providers. Also, Google allows hardware tokens as a second factor, so the account will remain secure. If a bank truly requires a cellular number for 2FA, I resort back to my instructions in the cellular services chapter.

As you have gone further through this book, you have likely seen many things become more about personal preference than global recommendations. Ultimately, your device should reflect your needs and desires. Various customizations allow you to create the perfect mobile device for your daily life. Please take the time to make everything perfect. It will help in the transition from phones which eavesdrop on our every move into your new private and secure device.

# CHAPTER THIRTEEN

## MAINTENANCE & TROUBLESHOOTING

If you currently possess an iPhone device, you are aware of the conveniences and overall simplicity it affords. Your iCloud account makes sure you lose no data and Apple's control over your device removes any complexities about your daily usage. GrapheneOS presents an environment in which you have total control without Google or Apple getting into your business. This freedom comes with responsibility. It is up to you to maintain your device's privacy and security. This chapter explains many considerations as you continue usage of this device.

### Updates

I believe the default update options within GrapheneOS are optimal. If you are switching over from a traditional Apple or Google device, I think you will be surprised at the frequency of security patches being pushed to your device. These are delivered via GrapheneOS, and not any third-party provider. About once every week or two, I am notified of a pending update. I allow the process to complete and reboot my device. The process is smooth and painless. You can work as normal during the download and installation. I could never go back to devices which deliver security updates a month or two after they have been published. This is another area where GrapheneOS excels. I have found no other mobile device operating system, including other custom un-Google options, which delivers updates as fast and as often as GrapheneOS. Call me a fanboy if you need to, I embrace it.

On rare occasion, I have updated the GrapheneOS operating system and reboot to find modifications to my settings. I have witnessed my mobile data connection become disabled, resulting in no internet access. If this happens, open "Settings" > "Network & Internet" > "Mobile Network", and enable "Mobile Data".

### Battery Drain

If you install GrapheneOS, and a suite of communication applications such as Signal, Molly, Wire, Tutanota, and others, expect some faster battery drain if you do not have push services enabled. Some apps may try to constantly listen for new incoming communications. This forces those apps to be ready at all times and prevents them from becoming dormant within the background. In my experience, this can change battery length with normal usage from two days to nine hours. Fortunately, there is a fix. The following is my process to regain proper battery life.

First, I charge the device to 100% and then use it as normal until the battery is almost dead. This usually takes me two days. I then monitor the battery history by navigating to "Settings" > "Battery" > "Battery usage". This will display all of the services and applications hitting your battery the most. If something stands out as inappropriate, begin your research. If you have a necessary application which is running in the background too much, consider the following change.

- Navigate to "Settings" > "Apps" > "See all...".
- Select your desired app and tap "App battery usage".
- Change the "optimized" setting to "Restricted".

Monitor how that app behaves after a full charge. After fully configuring my device based on the lessons in this book, I have not needed to make any changes, I consistently receive two days of usage on every charge. However, I turn my device off completely at night while I sleep.

## Backups

Once you have your GrapheneOS device configured, I encourage you to consider a backup. This will preserve most of your settings and customizations in case you need to rebuild your device. In previous versions of this guide, I recommended the native SeedVault backup application within GrapheneOS. Since the past few updates, it has become less reliable. Many readers reported errors or unfinished processes without any notification. Some reported the app would continuously attempt new backups on its own. Either way, let's move on to a better option.

The following command can be executed from within Terminal on macOS or Linux from the machine in which you previously installed ADB, as explained on page 12. Make sure your device is connected via USB cable and that debugging is enabled.

```
cd ~/Desktop
adb backup -all -system -apk -keyvalue -obb -shared -f backup.ab
```

The result will be a large file on your Desktop which should be a replica of the backup which would have been created locally on your device. make sure you store this somewhere secure with full-disk encryption. The following command would restore the backup to your device.

```
adb restore ~/Desktop/backup.ab
```

This solution is not perfect! I would only restore if absolutely needed, such as to fetch otherwise unavailable data. As I write this, developers from GrapheneOS have stated they plan to release a completely new way of conducting data backups. They seem to disapprove of SeedVault and encourage people to manually backup any vital data. Since I do not recommend keeping any sensitive documents on your mobile device, I don't worry too much about backups. They are mostly a convenience if you need to start over. If needed, you could use this book to rebuild your entire device in a day without backups. You will likely never need this backup, but it might save you hours of work if you lose your device or decide to upgrade to a new phone. I create a backup after configuration of everything mentioned in this book, but do not update it as time goes by. I confess I have never restored a SeedVault backup, as I have never had a device fail. When I get a new device, I embrace the opportunity to revisit every setting manually.



## Photos

If you are an iCloud or Google Photos user, you know that you never need to manually backup your photos. They are all conveniently sent to servers outside of your control just waiting for a breach. Transitioning to a private and secure lifestyle requires you to be responsible for your own content. The backup strategy I use, and encourage my clients to replicate, is as follows.

- Weekly, connect a FAT32 formatted USB-C flash drive to the mobile device.
- Open the Files application on the mobile device.
- Open the upper-left menu and select the device, such as "Pixel 6a".
- Navigate to "DCIM" > "Camera".
- Tap the three dots and choose "Select all".
- Tap the three dots and choose "Move to...".
- Open the upper-left menu and select the external drive.
- Tap "Move" in the lower area.
- Eject the external device and insert into a secure computer.
- Move the photos to your desired storage location.
- Make sure the photos were erased from the external device.

This is much less convenient than Google or iCloud storage, but it is also much more secure and private. By using the "Move" option instead of "Copy", I know the photos will be removed from the mobile device to make way for the next batch. I only store my photos, all of which I consider to be sensitive, within my personal laptop which possesses an encrypted drive. I then backup my entire laptop weekly to an external SSD drive. I do not store photos on my mobile device long-term. I view the mobile device as the camera and temporary storage.

## Notifications

While testing various settings within this book, I enabled push services and all notifications. My device seemed to continuously beep, flash, and buzz, which I found annoying. After disabling audible ringtones, I found all incoming calls still vibrated as an alert. I checked each app's notification settings and confirmed vibration was disabled. Yet, it still hummed every time a call came in. I found it helpful to also completely disable "Ring vibration", "Notification vibration", and "Media vibration" within the "Settings" > "Accessibility" > "Vibration & haptics" menu. I also did not completely disable "Ring & notification volume" under "Settings" > "Sound & vibration", as that re-enabled vibrating calls. Always take advantage of the search field within the Settings app to drill into settings which may be causing your own issues.

This chapter is much shorter than I expected, but I think that says a lot about the overall simplicity and useability of GrapheneOS.

# CHAPTER FOURTEEN

## DAILY USAGE & BEST PRACTICES

Your new private Android or iPhone may be all you need in regard to a mobile device. Most people carry it with them everywhere they go and leave it connected to the mobile network at all times. I believe this is risky behavior and a desire for extreme privacy will require you to take more extreme action. Some of my clients' primary mobile devices have never entered their homes and have never connected to a cellular tower within five miles of their houses. This prevents their phones from announcing their home locations. If someone did figure out a mobile number, and paid a bounty hunter or private investigator to locate a device, it would not lead anyone back to a home. The last known location should be a busy intersection with no connection to anyone. Some use a secondary mobile device with no cellular service within the home, which only connects via Wi-Fi. However, we should discuss whether you really need a secondary device.

I began presenting a secondary device option when I was still recommending Apple iOS devices. This was before GrapheneOS was available and I believed that Apple was our best option for privacy and security. This has changed. Apple is now collecting more information than ever before and continuously introducing new "features" which give us no control over their functionality. This presents a new dilemma for me. I had previously recommended Apple iPod Touch devices for use in the home while an iPhone could be used outside the home. While the Touch devices possess no cellular connectivity, they still collect and send data about you back to Apple every minute. Therefore, I am drastically changing my advice for secondary device usage. My new stance may not be well-received by some extremists.

We should first understand the reasons why any of this might matter to you. When you travel, your phone is always by your side and is your primary means of secure communications. When you return home, things might change. When you are about five miles away from home, at a very specific location, you might drop your device into a Faraday bag. This shielded pouch prevents any signals from reaching or leaving the phone. It stops all communications with cellular towers. The device might stay in this bag until you are at least five miles from home heading out on another trip. Since the phone is never connected to any network while near your home, it cannot reveal the overnight location of the device (or your home address). You might be surprised at the number of private and government organizations which have unlimited access to device location data.

While at home, you might still possess a secondary mobile device for secure communications. It might connect to your wireless network in the home (behind a firewall with VPN) and have internet access, but no activated cellular connectivity. Most secure communication apps, such as Wire, work the same as on your primary phone and can share accounts. Many use this strategy in order to possess a small device within the home without the need to rely on a large laptop all day.

One issue with this plan is the installation of Signal on the secondary device. Unlike username-based services such as Wire, Signal relies on a telephone number. Furthermore, it only allows usage on one mobile device at any given time. However, it provides a desktop application which can be used on multiple machines. Therefore, a secondary mobile device would not possess your primary Signal account, but your home laptop could. You can send and receive text, audio, and video over Signal while using a laptop.

This is where my new stance comes in. **For most GrapheneOS users who have the discipline to control their cellular, Bluetooth, and Wi-Fi connections, I no longer recommend a secondary device.** Please allow me to explain.

I insist on preventing any devices from connecting to any cellular network while in or near my home. These connections can immediately identify someone's location. When you place the GrapheneOS device into airplane mode, the cellular connection sends absolutely no data to any cell towers. The ability to block the microphone and cameras from the Quick menu further calms my worries. Unlike Apple devices, airplane mode is not disabled during updates or reboots. I simply trust GrapheneOS to maintain my desired connections more than Apple.

This option eliminates the secondary device completely, but would require some serious discipline. You could place the device into airplane mode while traveling and connect to Wi-Fi while at home. For extra credit (and comfort), you could remove the SIM before placing the device into airplane mode or disable the eSIM altogether. Since there is no Google or Apple account associated with the device, there is no central repository collecting data about the device's location and usage.

If you were to accidentally disable airplane mode, a connection would be made to a nearby cellular tower which could expose your location. However, who would know it is you? Your prepaid account is in an alias name and you never use that number for anything personal. The device was purchased with cash. The risk here is low, but there is still risk. I would never encourage a high-risk client to use their primary device in the home, but the majority of GrapheneOS users might have no issue with this. The pressure would be on you to enter airplane mode any time you are near your home. Only you can decide if this is feasible.

I followed the secondary iPod Touch strategy until 2021. Today, I do not use any iOS devices due to their requirement to possess a valid Apple ID, constant data collection, and increasing privacy invasions. What do I do now? **I have one GrapheneOS device.** It possesses everything I need. It only uses Wi-Fi while in my home. I enter airplane mode with disabled microphones and cameras when I am a few miles away from my residence. I disable Wi-Fi and airplane mode after I leave my house. I find myself relying on my laptop for the majority of my communications from home. I know many people who place their GrapheneOS device into airplane mode before they approach their homes and do not have issues of being tracked. Stock Apple and Android devices present greater risk. Ultimately, this all depends on your level of discipline and overall privacy and security threats.

## Faraday Bags

If you do not trust yourself to properly disable and enable connections from your device, you might consider a Faraday bag. These pouches block all signals into or out of a mobile device. I always have one available. I insist on all mobile devices to be properly secured inside a Faraday bag during sensitive meetings. I also require one when I travel.

I currently use the OUTPUT by Silent Pocket. My Pixel 6a fits tightly into the Faraday pocket, and the remaining wallet is protected from RFID signal leakage. This includes credit cards and identification. Furthermore, there are two money pockets which allows me to tuck away U.S. currency behind the Faraday pouch while foreign currency is readily available up front. The entire wallet zips completely to prevent any loose items from escaping. In warmer areas, I leave this wallet in my secure backpack due to the size, but it fits nicely into an interior jacket pocket when I am in cooler areas. This wallet ensures all of my most valuable items are together and secured from wireless sniffing while I travel.

Silent Pocket offers my audience a 10% discount when ordered through a dedicated affiliate link at <https://slnt.com/discount/IntelTechniques>, or using the discount code "IntelTechniques". The OUTPUT is pictured below. Note that the Pixel 7 and 7 Pro are too large for this wallet. The smaller 6a barely fits when not protected by a case, but is fully protected when properly inserted.



## Faraday Bag Testing

I insist on thoroughly testing any Faraday bags I purchase. Over the past ten years, I have acquired at least five bags which failed to prevent signals from entering or escaping the sleeve. Some may place their device in a bag, seal it, and call the phone number of the device to see whether it rings or forwards the call to voicemail. I do not believe this is an accurate test as you are relying on the signal strength of the nearest tower. A test in a rural area may be successful while that same test in an urban city could fail. Also, a failed call due to poor coverage may provide false assurances of the functionality of the bag. Instead, I rely on Bluetooth as my primary signal test. I can control the test better and apply strong local signals. The following is my routine with a \$15 small, portable, battery-operated Bluetooth speaker.

- Connect the mobile device to the speaker via Bluetooth.
- Play music from the device to the speaker.
- While music is playing, drop the mobile device into the bag and seal it.
- After the previous test, with music playing, drop the speaker into the bag and seal it while the mobile device is NOT in the bag.

In both scenarios, the audio should stop a few moments after sealing the bag. With some devices, the audio may play a while before stopping due to buffering. If the device continues to send multiple songs or a live audio stream to the speaker, then the bag is not performing appropriately. Now we should test other wireless signals.

- Connect the mobile device to Wi-Fi; stream an internet radio station from the mobile device through the internal speaker; drop the mobile device into the bag; and seal it. The audio should stop after any buffering of stored data.
- Disable Wi-Fi; enable a cellular data connection; stream an internet radio station from the mobile device through the internal speaker; drop the mobile device into the bag; and seal it. The audio should stop after any buffering of stored data.

In my experience, a poorly constructed Faraday bag is more likely to block cellular or Wi-Fi signals than nearby Bluetooth frequencies. I have yet to see a successful Bluetooth blocking test reveal that cellular frequencies were allowed. Therefore, Bluetooth is my baseline to detect the function of all Faraday bags. I also believe you should test the other connections as explained above. A Faraday bag should never be used before thorough testing. If your bag begins to show wear, repeat these tests. If your bag does not function properly 100% of the time, there is simply no point in using it at all.

## **Call Exit Strategy**

I offer an unorthodox telephone call strategy which may not be well-received with some readers. If you are ever on a call which becomes invasive, such as a company asking too many personal questions which you were not prepared to answer, never hang up the phone. This sends a message to the other party claiming the call was "ended" by you. Instead, place your device into airplane mode, including disabling of Wi-Fi. This will also end the call, but will send a message that the call "failed" but was not "ended" intentionally. You can later state that you had a service disruption without displaying the appearance of suspicious behavior. If you want to apply an extra dose of emphasis, disconnect the call while you are actively talking. If the other party calls you back, they will receive immediate voicemail instead of ringing without an answer.

## **Linux Phones**

In 2020, I saw the emergence of two privacy-respecting Linux telephones from Purism and Pine64. Both offer the ability to physically disable the cameras, microphones, and communications hardware. This alone is a huge feature for us. Both devices possess Linux operating systems which provide enhanced privacy and security. On the surface, these devices sound perfect. Unfortunately, this is not the case. Both devices rely on your cellular service provider for most standard calls and communication. VoIP is possible, but extremely limited. At the time of this writing, Wire, Signal, and Proton Mail do not fully support the operating systems. This eliminates the vast majority of features I require in a mobile device. I truly hope that the future presents a scenario where a Linux phone meets all of my needs. Until then, I do not recommend these devices. I believe GrapheneOS is far superior (and free).

## **Camera and Microphone Blocking**

Our mobile phones are designed to make life simple and fun. Most devices possess at least two cameras and numerous microphones. Selfies, high resolution photos, and speakerphone calls are simple thanks to the hardware present. However, these features can be used against us. Malicious software can enable a microphone or camera without our knowledge. In 2019, Facebook was caught secretly enabling the front camera of mobile devices while users were viewing their feeds within the app. Most social network apps circumvent security software by convincing you to authorize the necessary permissions to access your microphones and cameras. If you possess apps from Facebook, Amazon, and other providers, you will likely find that they all have unlimited access to your microphone and camera. Because of intentional and accidental exposure, I embrace both software and hardware camera and microphone blockers for the devices of all clients (and my own).

As previously mentioned, GrapheneOS has software-based disabling of all microphones and cameras. I trust this setting, but I do not trust myself. If I forget to disable the microphones after making a call or the cameras after taking a photo, I could be exposed. This is why I like to have a backup plan.



Camera blockers are easy. Much like a laptop, you can cover your mobile device cameras with black electrical tape or a dedicated removable sticker. Silent Pocket ([amzn.to/3twUUxq](https://amzn.to/3twUUxq)) offers reusable stickers designed to block embedded web cameras. They are more stable than generic options and are available in multiple sizes and colors. At a minimum, I encourage people to consider covering the front-facing "selfie" camera, as blocking the rear camera would also prevent any intentional photos. Due to paranoia, I keep both of my cameras covered until I need to use them. There are sliding plastic and metal products which easily enable the camera when desired, but I have found all of these to be poorly made and unreliable.

Microphone blocking can be tricky. Modern phones possess up to four unique microphones, none of which can be easily disabled. If a rogue app or virus began listening to your conversations, you would never know. The only fool-proof option would be to destroy each microphone, but that would make the device much less usable. Our best consideration is to "plug" the microphones. First, we must understand how microphones are chosen by system applications.

Think about your current mobile device. If you make a call and hold the phone up to your ear, you likely hear the other person through the small speaker near the top. The other party hears you through a microphone near the bottom. If you enable the speakerphone, you now hear the person through the speakers at the bottom. They hear you through the microphones at the bottom. Now imagine plugging in a set of earbuds with an in-line microphone. You now hear the other person through your earbuds and they hear you through the microphone within the cable. The operating system of the device detects all of this activity and adjusts the input and output based on your actions. Let's focus on that in-line microphone attached to your earbuds.

When you attach any type of headset which includes a microphone, your device detects this and switches the default microphone to the headset. It does not disable the other microphones. It only "listens" to the microphone which is plugged in. Now imagine if the microphone within the headset was broken. If you made a call through it, you would hear the other party, but they would not hear you. The device is only listening for the active microphone.

If you have an old set of earbuds you do not wish to use again, consider the following experiment. Cut the cable directly below the in-line microphone, but above where the cable splits for each ear. The remaining earbud will still work, but there is no microphone. The phone believes a microphone is present due to the plug structure. The phone enables the missing headphone microphone as the default and no one will be able to hear you on calls. This is the design behind a microphone plug.

Fortunately, you do not need to keep a pair of destroyed headphones plugged into your device in order to achieve these same results. Many companies offer "mic plugs" which virtually disable the working microphones of the device. The figure on the following page displays one of these options, a standard 3.5mm microphone plug made by Mic-Lock ([amzn.to/2B6QvXw](https://amzn.to/2B6QvXw)). This unit is larger than other flush-fitting models, but I have found it to be more reliable.



When you plug this device into your phone, it tells the operating system that you just inserted a pair of headphones with an in-line microphone. Therefore, your device makes this new mic the default option and tells all applications to listen to it if audio is needed. Since a microphone does not actually exist within this device, only silence is delivered. The Pixel 4a device I first used with GrapheneOS has a traditional headphone jack ready for these blockers. However, my 6a does not have this luxury.

Many newer mobile devices present a problem. Some do not possess traditional headphone jacks, and only offer a Lightning or USB-C connection. My 6a presents only a USB-C port. Mic-Lock makes Lightning and USB-C ([amzn.to/3v56Mso](https://amzn.to/3v56Mso)) plugs for these devices, as displayed on this page. There are numerous "L-Shaped" and miniature microphone blockers which are much smaller and fit flush to the device, but I avoid these for two reasons. First, many of these units unintentionally activate Siri or other apps because they send a virtual "long press" to the device. This causes battery drain and undesired Siri activations. Second, the smallest devices are often lost when removed. The larger plugs are easy to find and control. Also, their presence is obvious and you will know that you are protected.



Obviously, there are ways to defeat all of this protection. A truly malicious app or virus could be configured to ignore a headset microphone and force activation of internal mics. While possible, it is not very likely. I never consider these plugs to stop an extremely targeted attack. However, I believe they are valuable in blocking the common threats from social network apps and shady advertising practices. If you believe you would never be targeted for surreptitious video or microphone monitoring, consider the accidental "butt dial". Most of us have accidentally dialed someone from our mobile device while placing it into our pocket or a bag. That person can then answer the call and listen to us without our knowledge. A microphone blocker prevents this unintentional transmission of audio.

In 2021, a vulnerability with numerous communications applications, including Signal, was patched after a security researcher reported his findings. A call could be placed to a mobile device along with a malicious command which instructed the recipient's device to automatically answer the call. This would have allowed the intruder to listen to you at any time without your knowledge. While this specific issue has been fixed, we all patiently wait for the next problem. A microphone blocking device would have prevented this attack from successfully monitoring your conversations. The moment I end an audio call on my mobile device, I insert the mic blocker into the headphone port. This way I know that I can no longer be heard. I do not trust the tap of a virtual

button on a piece of glass to properly inform the software to end the call. Have you ever participated in a group FaceTime call or conference chat and accidentally pressed the option to activate your device camera? Have you ever accidentally un-muted yourself during mandatory company group calls? I know I have done both. Fortunately, my camera blocker stopped any video transmission to the other participants and my microphone blocker prevented an embarrassing moment.

Hopefully, you will never need to rely on the protection of these physical blockers. If you are diligent about disabling microphones and cameras from your software, you need none of this. If you are like me, you enjoy a physical representation of your additional layer of protection. Proper execution eliminates threats and provides peace of mind. It is now habit with me to also disable microphones, cameras, and location when I enable airplane mode from the Quick Menu. After a while, it should become second nature.

### **Wi-Fi & Bluetooth Tracking**

There is a new trend in customer tracking which concerns me. Many retail stores, shopping malls, and outlet centers have adopted various wireless network monitoring technologies in order to follow customers throughout a shopping area. These rely on your Wi-Fi and Bluetooth emissions from your mobile device. When you enter a store, your signals are collected and stored. As you move around, various sensors attempt to identify your exact location and length of time within a specific area of a store. If you leave without purchasing any items, you might be tracked by the neighboring store and your pattern is helpful to their customer analytics. This may sound too futuristic, but it happens every day. Random spoofing features being adopted by Apple and Android help with this invasion, but companies always find new ways to track us via the signals our devices broadcast at all times.

My solution to this is simple. The Bluetooth and Wi-Fi signals on my travel phone are always off. Many will resist this, as keeping these connections enabled is very convenient. Your device will immediately connect to your car stereo and switch over to your work Wi-Fi when you enter the building. However, this comes at great risk.

If I want to connect my device to my car stereo in order to listen to music or a podcast, I rely on a physical audio cable. I do not recommend connection via a USB cable within vehicles which offer a USB port into the entertainment system. This can be abused if your vehicle collects device details and transmits them over a cellular data connection. Instead, I insist on a standard audio cable which plugs into the 3.55 mm stereo port available in most modern cars. If you have a modern Pixel, you will need a USB-C to 3.55 mm adapter, as explained next. Once you have a device which is capable of this connection, rely on a standard 3.55 mm male to male stereo audio cable without requiring any wireless signals or USB connections. Please eliminate technologies which make you easier to track.

## Headphones & In-Line Microphones

I confess I am a bit of an audio snob. I miss the old days when mobile devices possessed a decent Digital Audio Decoder (DAC) which piped music to a traditional 3.5 mm plug, ready for any common set of headphones. Today, finding this feature on any modern device is surprising. As already stated, modern Pixel devices only provide a USB-C plug at the bottom. This presents a problem.

I insist to never communicate during an audio or video call through the device's speakers. This includes the traditional earpiece or the speakerphone option. I believe this is rude to the other end of the call and the people around you. I do not want anyone listening to the details of my conversation and I don't want to annoy others around me. Therefore, I always use an in-ear headset with an in-line microphone. This allows me to talk at a controlled level and hear the other end through both ears at a lower volume. It stops most of the other end of the conversation from audibly leaking out to the public around me.

The solution most people apply is Bluetooth headsets. These are out for me as I never activate my Bluetooth connection. Some companies are starting to make USB-C wired headsets, but this is not wide-spread yet. Therefore, you will probably need a USB-C DAC adapter. I have two recommendations, depending on your desired audio quality.

A cheap converter, such as the \$9.00 Apple USB-C to 3.5 mm Headphone Jack Adapter (<https://amzn.to/3K1BhJw>), will work for most people. It allows you to plug in your existing headset or earbuds and sounds fine. If you typically need to crank the volume on calls, you may be disappointed in this. It is not very loud, even at full volume. I currently use it for daily calls.

A better DAC is the \$50.00 FiiO KA1 (<https://amzn.to/3XnAHJ4>). I could hear the call at a very loud volume, as this device possesses a true amplifier. However, I had to speak into the microphone of my Pixel in order to transmit audio. I was never able to make it accept an in-line microphone from a connected set of earbuds. This was not a huge inconvenience. I currently use this small device to power my in-ear monitors while listening to music. It amplifies a clean sound, and is much better than the cheap DACs. If you apply EQ to your music, I believe this is a valuable appliance. If you listen to online music streams at a low volume with cheap earbuds, it will do nothing for you.

## Data Transfer

Since we do not have native access to Google or Apple cloud-based storage, we are responsible for our own data. This means you will likely need to copy data out of your mobile device and onto a computer. Most commonly, you may need to export your photos this way. There are many programs which allow you to connect your device to your computer via USB-C cable, but I find these slow and annoying. Instead, I dedicate a USB-C external flash drive for this purpose. I currently recommend the SanDisk Ultra Dual Drive line (<https://amzn.to/40SIOjQ>). I have learned my lesson buying cheap drives. These are fast and reliable. An image is present on the following page.

Upon opening the product, I placed it in my Pixel and allowed the GrapheneOS system to format the drive as desired. From there, I can copy data to it through the USB-C connection and then export that data to any computer through either a USB-C or USB-A port. I find this much more reliable than transferring data via USB cable with the mobile device in file transfer mode.



### **Decoy Phone**

I have been carrying a secondary phone during travel for over a decade. This began as a Wi-Fi device which did not possess a SIM card or cellular service. I used VoIP options such as Google Voice to make calls without any connection to my primary device, which was a government-issued Blackberry at the time. One day, I dropped this device and shattered the screen. I needed to make a personal call while in a meeting at a hotel. I walked to the front desk, showed the receptionist my phone, and asked if I could use the hotel phone. She obliged without any hesitation, and even offered her sympathy to my situation and need to purchase a new device. This ignited a spark in my brain.

Today, I keep a small, lightweight, and severely outdated Android device with a cracked screen in my backpack at all times. I removed the battery to eliminate further weight. The following explains a few usage scenarios I have found beneficial. I am confident you will find others.

- During the COVID-19 pandemic, I found many restaurants which only offered carry-out services and no inside dining. These businesses required patrons to download invasive apps to place orders and retrieve the food. Many required scanning of QR codes which then prompted download of questionable software. Polite requests to pay with cash and avoid the apps were denied. However, displaying my broken phone magically presented an option to order food without sharing my personal details.
- While in a library using public Wi-Fi in order to create anonymous online shopping accounts, I needed to attach and confirm a telephone number with my account. I explained to a staff member that I had broken my phone (while holding the device in obvious view) and asked if I could receive a confirmation code through one of their telephones. She happily allowed me to use a fax machine to receive the call and obtain the code.
- While seeking chiropractic care with a new provider, I was told I had to enter a cell number into their system for text-based appointment reminders. This was mandatory for all patients and any data collected was shared with third

parties. I sadly displayed my broken device and asked if I could provide these details on the next visit after I activated a new device. This was allowed and I was never asked again.

I often see mobile devices with cracked screens for sale on Swappa, eBay, and Craigslist. You may have an old device which can be dropped a few times until the desired result is achieved. If you do not want to carry two devices or have no desire to break your own phones, you might consider a "cracked screen" application. These apps create a simulation of a cracked screen. They are not always convincing, but should work from a distance.

### **911 Phone**

You may now have the perfect mobile communications configuration with an anonymous device and service with VoIP calling options to protect your true number. What will you do if there is an emergency? If you call 911 from your device, your true number will be captured and documented. If the police contact you, your name and other details may be added to a public report. I encourage you to think about this now and have a plan. If you have a true emergency and only have your primary device to call 911, do it. Your health and safety are more important than anonymity. You can always buy a new SIM later. However, I keep a "911 phone" in my vehicle at all times, along with a power cable. Mine is an old Motorola flip phone. It has no SIM card or account details. Any functioning cell phone will allow a call to 911 through the closest tower without any activation.

### **Cases**

A protective case for any mobile device is a personal choice. There are those who always place their mobile device inside some type of rubber or plastic case to prevent slippage and minimize drop damage, and the daring who leave it naked. I prefer the Spigen Thin Fit (<https://amzn.to/3RTmOB6>) for my 6a.

### **Summary**

Hopefully, you now possess a new phone with absolutely no public connection to you. It has service through a prepaid provider which does not know your true identity. The service is paid through either prepaid or masked cards. The phone has never connected to any cell towers near your residence thanks to your new Faraday bag or software control strategies. Nefarious apps cannot take complete control of your device. There is no cellular location history associated with your home.

# CHAPTER FIFTEEN

## RESET & REVERSAL

You may have experimented with GrapheneOS and decided it was not ideal for you. You may have regret and wished you could just return to the stock operating system as it was the day your device was purchased. You might want to sell your device but need to revert the settings and reinstall a stock Android build for the next user. Fortunately, this is quite easy. The following steps erase all personal data and restore the device to the original Android operating system which would be present if purchased as a new device.

### Browser-Based Solution

The easiest way to reset your device to stock Android is through Google's official installation site. Much like the GrapheneOS installation option, a Chrome-based browser is recommended. During my testing for this book, I downloaded Brave Browser; completed all installations; then uninstalled Brave Browser. Conduct the following to restore stock Android to your device, which will erase all data.

- Tap "Settings" then "About phone".
- Tap "Build number" seven times to enable Developer Options.
- Go back one screen and tap "System".
- Tap "Developer options" and make sure it is enabled.
- Enable "OEM unlocking" from this screen.
- Enable "USB debugging" from this screen.
- Connect the device via USB cable to your computer.
- Allow all pop-ups within your desktop browser settings.
- Navigate to <https://flash.android.com/welcome> from your desktop.
- Click "Get Started" or any other welcome screen (which changes often).
- If prompted to install drivers, choose "Already installed".
- Confirm option to allow ADB.
- Confirm USB Debugging access on the mobile device.
- Select your device from the browser site.
- Select your desired build, such as "Back to Public".
- Confirm "Wipe Device" and "Lock Bootloader" are enabled.
- Click "Install Build", "Confirm", and "I Accept".
- Allow the operations to complete.
- When prompted, reselect your device and allow the connection.
- Allow the process to complete.

Your device should reboot into stock Android. Please note that the exact steps for this process could change slightly, but the website should walk you through each step.

## Computer-Based Solution

A computer is required for this process, and I only provide the steps for an Ubuntu Linux system. As long as you have "ADB" installed on a macOS or Windows computer, all of these steps should function for you. You would only need to ignore the Linux ADB installation options.

- Navigate to <https://developers.google.com/android/images>.
- If required, click "Acknowledge" at the bottom.
- Identify the "images" for your specific hardware (mine was "bluejay for 6a").
- Download the most recent version by clicking the "Link" option next to it.
- Power the mobile device on.
- Tap "Settings" then "About phone".
- Tap "Build number" seven times to enable Developer Options.
- Go back one screen and tap "System".
- Tap "Developer options" and make sure it is enabled.
- Enable "OEM unlocking" from this screen.
- Enable "USB debugging" from this screen.
- Connect the device via USB cable to your computer.
- Open Terminal (or Command Prompt) and execute the following.  

```
sudo apt install android-sdk-platform-tools-common
adb reboot bootloader
fastboot flashing unlock
```
- Press the volume button on the device to select "Unlock Bootloader".
- Press the power button on the device to execute the selection.
- Decompress (unzip) the downloaded Android file on your computer.
- Within Terminal, navigate to the folder of the unzipped file (or type `cd` and drag the folder from Files into Terminal).
- Within Terminal, type `./flash-all.sh` and press enter.
- Allow several processes to finish and the device to reboot completely.
- Skip all prompts.
- Enable "Developer options" as previously explained.
- Enable "USB Debugging" as previously explained.
- Within Terminal, execute `adb reboot bootloader`
- Within Terminal, execute `fastboot flashing lock`
- Press the volume button on the device to select "Lock the bootloader".
- Press the power button on the device to execute the selection.
- Unplug the device and reboot.
- Confirm that all Developer Options are disabled.

Your device should now boot to a standard Android operating system and your bootloader should be locked. There will no longer be a warning upon powering the device on about a custom operating system. All personal data and custom settings have been erased and the device is safe to issue to someone else.



# CHAPTER SIXTEEN

## APPLE iOS CONSIDERATIONS

I believe the privacy and security of a custom un-Googled Android device is far superior to any stock Apple or Android phone available from retail stores. Unfortunately, my clients are usually most familiar with the iOS environment and some demand these devices. Therefore, I am always ready to meet these expectations. If I cannot convince a client to switch to GrapheneOS, I typically purchase iOS phones with cash at an Apple store and leave without accepting Apple's activation and setup services. If you purchase a device online, there will always be a digital trail to your true identity. Therefore, cash in-person is always preferred.

Once you have your new device, you are ready to configure all settings and create an Apple ID account. There is a lot to consider. If you purchased a new device, this is a great opportunity to establish a new Apple ID and prepaid cellular account in order to stop the tracking of your old accounts and restart the data collection process with anonymous details. Conduct the following on your new device, which is based on iOS 16.3. Future versions may appear slightly different.

- Turn on device.
- Select language and region, then click "Set Up Manually".
- Select and join available Wi-Fi.
- Click "Continue".
- Set up Touch ID if desired.
- Click "Passcode Options" and choose "Custom Numeric Code".
- Create a strong passcode and click "Next".
- Confirm passcode and click "Next".
- Choose "Don't transfer data and apps".
- Click "Forgot password or don't have an Apple ID".
- Choose "Set Up later" in Settings.
- Choose "Don't Use".
- Agree to the terms of service.
- Click "Continue" or "Customize Settings".
- Choose "Not Now" for "iMessage and Facetime".
- Choose "Disable Location Services".
- Choose "Setup Later in Settings" (Siri).
- Choose "Setup Later in Settings" (Screen Time).
- Click "Don't Share" iPhone Analytics.
- Select desired appearance and zoom.
- Click "Get Started" to exit the menu.

Once you have booted to the main menu, the following configurations should be considered through the Settings menu. Note that some of these settings may disable features which you find desirable, and some options here might not be present within your device. Research any modifications and apply settings which are most appropriate for your usage.

- Settings > Wi-Fi: Off (If not used)
- Settings > Bluetooth: Off (If not used)
- Settings > Cellular: Disable access to undesired apps (Find My, Contacts, etc.)
- If using ONLY cellular data, and not Wi-Fi, you can use this menu as a firewall.
- Settings > Notifications > Scheduled Summary: Off
- Settings > Notifications > Show previews: Never
- Settings > Notifications > Screen Sharing: Off
- Settings > Notifications > Siri Suggestions: Disable all
- Settings > Notifications: Disable notifications on sensitive apps
- Settings > Notifications: If desired, disable all Government Alerts
- Settings > General > AirDrop: Receiving Off
- Settings > General > AirPlay & Handoff: Disable all
- Settings > General > Picture in Picture: Disable
- Settings > Siri & Search: Disable all
- Settings > Siri & Search > (each app): Disable all
- Settings > Privacy & Security > Location services: Disable all
- Settings > Privacy & Security > Tracking: Disable all
- Settings > Privacy & Security > Research Sensor & Usage Data: Disable all
- Settings > Privacy & Security > Motion & Fitness: Disable all
- Settings > Privacy & Security > Analytics & Improvements: Disable all
- Settings > Privacy & Security > Advertising > Personalized Ads: Disable
- Settings > App Store > Video Autoplay: Off
- Settings > App Store > In-App Ratings & Reviews: Disable
- Settings > Passwords > Security Recommendations > Detect...: Disable
- Settings > Messages > iMessage: Disable
- Settings > Messages > Share Name and Photo: Disable
- Settings > Messages > Shared with You: Disable
- Settings > Messages > Show Contact Photos: Disable
- Settings > Facetime > Facetime: Disable
- Settings > Safari > Siri & Search: Disable All
- Settings > Safari > Search Engine: DuckDuckGo
- Settings > Safari > Search Engine Suggestions: Disable
- Settings > Safari > Safari Suggestions: Disable
- Settings > Safari > Quick Website Search: Disable
- Settings > Safari > Preload Top Hit: Disable
- Settings > Safari > AutoFill: Disable All
- Settings > Safari > Prevent Cross-Site Tracking: Enabled

- Settings > Safari > Fraudulent Website Warning: Disable
- Settings > Safari > Privacy Preserving Ad Measurement: Disable
- Settings > Safari > Check for Apple Pay: Disable
- Settings > Safari > Camera: Deny
- Settings > Safari > Microphone: Deny
- Settings > Safari > Location: Deny
- Settings > Maps > Share ETA: Disable
- Settings > Maps > Air Quality Index: Disable
- Settings > Maps > Weather Conditions: Disable
- Settings > Maps > Ratings and Photos: Disable
- Settings > Maps > Show Ratings and Photos Suggestion: Disable
- Settings > Maps > Follow Up by Email: Disable
- Settings > Shortcuts > iCloud Sync: Disable
- Settings > Shortcuts > Private Sharing: Disable
- Settings > Music > Show Apple Music: Disable
- Settings > Camera > Scan QR Codes: Disable

Remove any unwanted optional stock apps, such as Home, Translate, Books, iTunes Store, Watch, Tips, Facetime, Calendar, Mail, Notes, Reminders, News, TV, Stocks, etc. Change the wallpaper if desired and remove unwanted Widgets from screens. Remove any unwanted apps from home screen and create new app shortcuts if desired.

You should now have an iPhone with several custom configurations. However, you have not connected an Apple ID to your device yet. You cannot download any apps. I like to establish a new Apple ID at least once a year in order to slightly confuse Apple's data collection systems. I insist on a new Apple ID and prepaid cellular account any time I switch to a new device. Beginning with iOS 15, signing in through the standard Apple ID menu logs you into iCloud without an option to disable overall synchronization (you can only disable individual services). This is dangerous, especially after rebooting during an update. The following should bypass the mandatory iCloud option and establish a new Apple ID for your device.

- Open "App Store".
- Click "Continue".
- Click "Turn off Personalized Ads".
- Click Profile icon in upper right.
- Click "Create New Apple ID".
- Provide (and document) desired Email and Password.
- Provide (and document) desired Name and DOB.
- Disable App Updates.
- If prompted, choose "None" as payment type.
- Insert desired Street number, City, State, and Zip.
- Provide desired number and verify via text (discussed next).
- Verify email confirmation code if prompted.

I was able to create an account by providing a name, new email address, and VoIP number (Google Voice) while connected to a VPN. Your mileage will vary here. Apple may block VoIP numbers during your registration. This brings us to a major deviation from previous editions. At one time, I insisted on preventing Apple from knowing my true cellular telephone number and never shared it with my Apple ID account. We now know that Apple continuously collects hardware identifiers, such as a serial number and telephone number associated with the SIM card inside the device. Therefore, Apple knows your cell number and a unique device ID.

Because of this, I see no real reason to hide your (anonymous prepaid) number from Apple. This also eliminates the need to present Apple with a VoIP number. I place the activated SIM within the iOS device, confirm I can receive text messages, and provide that number to Apple during the Apple ID account creation process via the App Store.

In 2021 and 2022, I was able to activate Mint Mobile SIM cards by calling their support or chatting through their online site. I explained my dilemma of not being able to download the app without an Apple ID; not being able to create an Apple ID without giving them my number; and not receiving my new number without the app. I then provided all details from the card; inserted the card into the new iOS device; identified the desired area code for service; waited for them to activate and send a test message to my new number; and provided that number during the Apple ID registration process. I no longer need to maintain a VoIP number just for this purpose. If you are registering new anonymous prepaid service within a new iOS device, I believe providing that new telephone number during the Apple ID creation process is the best option today.

We can now continue removing some annoyances with the following steps.

- Open Settings and notice your new Apple ID is present.
- Click "Finish Setting Up Your iPhone" then "Finish Setting Up".
- If prompted, enter your PIN.
- Click "Cancel" then "Back".
- If present, click "Start using iCloud", then "Not Now".
- Click your Apple ID account and confirm iCloud is "Off".

After you have logged into your new Apple ID account, you may notice that Apple enabled iCloud without your consent. This is common behavior and their attempt to lock you into their cloud storage options. This can be corrected quite easily. Navigate to "Settings" and click on your new Apple ID account. If the "iCloud" option displays "Off", there is nothing you need to do. If it displays anything else, then you are logged into iCloud and Apple is collecting data about you and your device. From this screen, choose the "Logout" option and allow your device to remove data from iCloud. Return to the App Store and log in to your new account. This should present a banner notification under "Settings" to complete your setup. Tap this and choose "Not now" for iCloud. Confirm the iCloud setting displays "Off". This setting should stay in place as long as you take no action to enable iCloud.

We have just a few more customizations. Consider the following.

- Settings > Apple ID > Media & Purchases > View Account > Recommendations: Disable
- Settings > Apple ID > Rating & Reviews: Remove All
- Settings > Apple ID > iCloud: Disable all
- Settings > Apple ID > Find My > Share My Location: Disable
- Settings > Finish Setting Up Your Phone > Set Up Siri: Setup Later in Settings

After you have successfully signed into the App Store, but not iCloud, open the Find My app and confirm that it prompts you to sign into an account (but do not sign in). This indicates that some basic abilities to track your device are disabled.

### **iOS Application Considerations**

App selection and configuration is a very personal choice. I present my preferences here which may vary compared to yours.

**Lockdown Privacy:** This firewall blocks most analytics, trackers, and malicious connections being made within apps and the operating system. Default configuration is minimal, so I tap "Block List", then enable each category option. If you apply my DNS filtering technique in the previous text, you may not need this app. Note that Apple bypasses these types of apps to make their own connections. Therefore, I believe this app is only helping slightly today, and that DNS filtering is superior.

**Firefox Focus:** This is my preferred iOS default browser, and I modify the following.

- Settings > Firefox Focus > Default Browser App: Firefox Focus
- Firefox Focus > Settings > Safari: Enable
- Firefox Focus > Settings > Send Usage Data: Disable
- Firefox Focus > Settings > Search Engine: DuckDuckGo
- Firefox Focus > Settings > Get Search Suggestions: Disable

**MySudo:** MySudo users should make sure they have access to their account within a current iOS device if you plan on transferring it to another "new" device. You can then export the account into the new iOS device by going to Settings > Backup & Import/Export > Export To Another Device. If you are resetting your only iOS device, you would need to restore from a good backup. If this is the case, there is little reason to reset your device. You can still create a new Apple ID within your updated device without losing the MySudo accounts currently present. Users should consider all of this and plan appropriately if making any changes. I modify the following.

- MySudo > Settings > Privacy: Disable all
- MySudo > Settings > Team Sudo Updates: Disable all
- MySudo > Settings > Backup & Import/Export: Set a Recovery Password
- Backup the device locally

Proton Mail: Regardless of your primary settings, the Proton Mail iPhone app will add a footer at the end of every email announcing your usage on a mobile iOS device and set your default browser to Safari. I take the following actions:

- Proton Mail > Settings > Account > Mobile Signature > Disable
- Proton Mail > Settings > Default Browser: Firefox Focus

Proton VPN: If you plan to use both Proton VPN (or any other VPN) and Lockdown Privacy simultaneously, you must change the protocol of your VPN. Go to your settings and disable Smart Protocol. This allows you to change your protocol to IKEv2 which eliminates the conflict with Lockdown Privacy. However, I again encourage you to seek a DNS filtering replacement for Lockdown.

Signal: I modify the following.

- Signal > Settings > Privacy > Default Timer: Set as desired
- Signal > Settings > Privacy > Hide Screen in App Switcher: Enable
- Signal > Settings > Privacy > Show Calls in Recents: Disable

Strongbox: This application opens KeePassXC databases. I prefer to keep my mobile version of passwords "Read Only" and only make changes from my laptop when necessary. The biometric option to open databases is available with the paid version.

Backup and Restoration: Backing up your iPhone is much easier than Android. It only requires you to open Finder on your new Apple computer with Catalina or later operating system, connect the mobile device via USB, and conduct the following.

- Click the phone option in the left menu.
- Scroll down and click the "Back Up Now" button.

This will create a backup of the operating system configuration and all Apple data such as your contacts, notes, and calendars. It does not backup all apps and their settings or any media such as music. If you do not possess an Apple computer, you could use iTunes installed to a Windows machine. If you want extreme privacy, you could set up a Windows virtual machine on a Linux host; disable all internet access to the Windows VM; install iTunes within the Windows VM; and connect your mobile device to the iTunes installation. Regardless of the way you do this, having a backup of your mobile device settings will be a huge benefit if you ever need to replicate your configuration onto a second device. This is vital for my clients, as I will not be with them when a disaster happens.

For extreme privacy, this device should never be configured from your home. Most phones have location services, Wi-Fi, Bluetooth, and cellular connectivity enabled by default. This could expose your account and associate it with your residence. Also, most Apple updates re-enable all radio connections.

If you plan to purchase apps, obtain a prepaid iTunes gift card with cash from a grocery store. Never provide Apple with a credit or debit card number. Hopefully, this will not be necessary because you should possess minimal applications and only those absolutely required.

For most clients who demand an iPhone, I encourage them to obtain the latest generation iPhone SE. This device has plenty of power and is quite affordable. The main feature I like is the fingerprint sensor. While I do not use it, I know my clients do. I would rather them apply a fingerprint to unlock the device instead of the default facial recognition included with flagship iPhone models.

Regardless of the model, I immediately disable all iCloud services within the device. This will prevent accidental exposure such as emails, contacts, calendars, and notes from being stored within Apple's cloud storage. While I do not recommend using Apple's stock iOS applications for any of these services, it is easy to upload data unintentionally. You can access these settings from the iOS "Settings" app > "Apple Account" > "iCloud". This should display "Off" within this menu. Hopefully, you were never signed in.

Some may question my distrust of iCloud. A more appropriate claim would be that I do not trust any cloud storage services for my clients. We have all heard about various breaches which exposed celebrities' personal photos and email messages. These occurred due to the convenience of free cloud storage. The only way to truly prevent this is to block any data from leaving the device. Most of my clients are highly targeted due to their fame, so I insist on completely disabling iCloud or any other storage solution.

Many people ask about the security of the Touch ID option. I do believe it is secure, and Apple does not receive an image of your fingerprint. Your device creates a mathematical value based on the print, and only looks for a match when it is used. It is only as secure as your passcode, since either can unlock the device. Your decision to activate Touch ID is personal, and most of my clients demand it. I only ask you to consider the following threats.

- **Forced Print:** If you are placed under physical duress, you could be forced to use your finger to unlock a device. This is extremely rare, but I have had clients who were victims of kidnapping and abduction. These unfortunate incidents weigh heavily on this decision.
- **Legal Demands:** Some courts have ruled that providing a passcode is not always required as part of a search warrant to search a device, but a fingerprint is. You can refuse to tell your code, but may be physically forced to give up your fingerprint.
- **Apple Face ID:** I would never consider using this. Although Apple does not store your image, it has been proven vulnerable using photographs of faces to unlock the device.



As I stated previously, I never use cloud storage for sensitive information such as personal photos and videos. However, I respect the need to possess a backup of this data, especially when our mobile devices likely create and store every image we capture. Since many clients possess a new iPhone and Apple computer, I encourage them to manually backup all content via USB cable. The default Apple application for photo backups is Photos, but I prefer not to use it. Instead, I use the stock application titled Image Capture. This minimal software does not attempt to connect to Apple servers and has limited functionality. Upon connecting an iPhone to an Apple computer, I conduct the following.

- Launch Image Capture and select the iPhone in the upper right.
- In the "Import To" option, select the folder which will store all images.
- Select "Import All" to copy all images and videos to the computer.
- If desired, select all images, right-click, and permanently delete from the device.

If you are frustrated at the requirement to use Apple's iTunes or Music app to transfer music to your device, I have eliminated many of the headaches by using a premium application called iMazing. It allows me to transfer music, photos, contacts, documents, and backups to or from any iOS device without complications from Apple. The ability to transfer new music files without the possibility of deleting all stored songs is worth the \$45 price to me. If you have this software, you do not need any stock apps from Apple in order to import or export any type of data associated with your mobile device.

Once you have your photos and videos on your computer, I hope you are conducting backups of your data to an external device. By maintaining all of your personal data locally on machines in your possession, you completely eliminate the ability to "hack" into your iCloud and steal your content. You are not bulletproof, but an attack would be extremely targeted and difficult. Note that connecting your new iPhone to your new Apple computer creates a known connection of these two devices with Apple. The risks are minimal since both devices hopefully have no association to your true identity.

If you do need to rely on iCloud storage, please execute two important modifications within your iCloud settings. First, disable web-based iCloud access. This prevents someone from using your credentials to access your iCloud account via a web browser. Only your devices will be able to gain access. This could stop some common exposed password vulnerabilities. Next, switch to a more secure 2FA, such as a YubiKey, which is supported as of early 2023.

I want to state again that I do not use iOS devices and never recommend them to people able to transition to a GrapheneOS device. As I was updating this chapter, I fetched an iPhone SE from my collection of retired products to test all of these settings. Halfway through the steps, while simply trying to download a free app, I was blocked by Apple. They wanted my password again, which I provided. They then demanded that I verify my VoIP number on file, which I did. They then required a code be sent to that number, but then refused to send the code. They also refused to

allow me to receive a code at the email address on file. Everything I tried to get access to my own account failed. Apple suspended me from access to anything in their app store, all because I wanted a free app. This summarizes my dislike of Apple and relief to have found a more reliable operating system (GrapheneOS).

If you were blocked from accessing any new apps or the content within iCloud, would you be impacted? I have countless stories of being on the road and having limited functionality within my mobile iOS device. This is why I always prefer to use devices which do not mandate an active online account in order to receive full access to the device. Apple has the power to lock you out at any time. Customer support will not help you when this happens unless you can pass all scrutiny. If you adopt various privacy practices, Apple will not like this and refuse to assist.

My final thought within this chapter comes directly from my experience with numerous celebrity clients and the online attacks which forced them to retain my services. They all had iPhones with active iCloud accounts. Their data was automatically synchronized in the background. When online criminals gained access to those accounts due to password recycling or other behaviors, they had everything needed to steal, extort, and harass my clients. The best defense against this activity is to never synchronize the data online. If your photos never leave your devices, there is no easy way to access the data. This is a vital step to extreme privacy if you choose to use Apple devices.

I have bashed Apple a lot in this chapter. However, I do believe their operating system is secure. I believe their intention is to make the iOS experience easy and convenient for their users while offering a decent sense of privacy on the surface. However, Apple wants to know everything about you through default settings. If you modify your settings, disable iCloud, create an anonymous Apple ID, and use a prepaid account, I believe your privacy risk from iOS is less than the traditional user.

# CONCLUSION

I hope you now possess a private and secure mobile device which does not share all of your activity with Google or Apple. Once you leave those invasive ecosystems, I believe you will find the minimalism and simplicity of your new device to be a superior experience. I practice what I preach, and configured my own personal device from this guide. I could never go back to any Apple iOS or stock Android device now. I no longer worry about unnecessary data collection. My phone no longer feels "dirty" a few weeks after using it. There is a great sense of freedom when you leave that world behind.

If this document should need updated, all modifications are completely free. If you purchased this PDF through my website, you will be notified via email when revisions can be downloaded. If you downloaded an unauthorized copy from a book piracy website, please consider purchasing a legitimate copy. Your \$15 purchase supports my ad-free podcast and all of the research which goes into creating and updating these guides.

Thank you for the continued interest in Privacy, Security, & OSINT.

~MB

IntelTechniques.com