EXTREME PRIVACY: macOS DEVICES

MICHAEL BAZZELL

DIGITAL EDITION

Order 0145178083

EXTREME PRIVACY:

macOS DEVICES

MICHAEL BAZZELL

EXTREME PRIVACY:
macOS DEVICES

First Published: May 2023

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

Cover Concept: Anonymous Podcast Listener

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 May 19, 2023. Readers may find slight discrepancies within the methods as technology changes.

Revision: 2024.07.01

CONTENTS

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 reader to find any exact terms or phrases, as that is much more beneficial than any index or hyperlinked table of contents.

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 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*.

macOS 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 overwhelming at the time. We threw 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 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, and the previous ***Mobile Devices*** eBook, 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 efforts. 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. 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. If you bought this, thank you for your support! If you did not, consider purchasing a legitimate copy in order to receive all future updates. If you find anything which needs updated or corrected, please email us at books@inteltechniques.com. My staff cannot respond to emails directly, but they will monitor them for any changes which we need to apply to the next version of this guide.

With ***Extreme Privacy: macOS 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 macOS devices.

INTRODUCTION

In 2019, I requested all of the data stored about me from Apple. I was preparing to record a podcast about the data which Apple collects about all of us, and I wanted to see the damage first-hand. I possessed several Apple ID accounts in alias names, so I requested the data from each of them. Since I never gave Apple my true name, I assumed the exposure was no big deal. I was wrong.

First, I received the data associated with my MacBook Pro purchased in 2015, which I had not used since an upgrade in 2018. The download included many files, and the amount of data was overwhelming. I opened the files related to my Apple ID account and learned that Apple knew the following, which I found to be harmless.

- My full alias name
- My alias email address provided during account creation
- My alias physical address provided during account creation

I felt good about this. Apple did not know my true name. I did well with my privacy. Then I dug deeper. Apple also knew the following.

- The email address was also associated with another MacBook Pro
- The serial number for both devices
- The date I first used the email address with Apple
- Multiple IP addresses possessed during use of the devices
- The internal computer names assigned to all devices
- The dates/times of any reformatting of the OS
- The dates/times and IP address of last access to iTunes, FaceTime, and iCloud
- My time zone during usage of the devices
- The VoIP telephone number provided during an Apple account login in 2017

This was disturbing, but Apple still did not know my true name, address or personal activity. I then began going through the additional files. Apple also knew the following.

- An alternative email address I once entered into Apple Mail
- Songs I had listened to through the official Apple Music application
- The moment within the songs when I paused the playback
- My IP address during media streaming from Apple's servers
- My preferred musical artists identified during their onboarding process
- The serial number of an iPad which I had accessed on 04/08/2012
- An alias birth date provided on the iPad
- The alias name provided for the iPad
- The serial number of an iPhone which I had accessed in 2017
- All podcasts subscribed to through the iOS device

- Titles of podcast episodes which had been completed or paused (hundreds)
- Dates of podcast subscriptions and listening times
- Podcasts which possessed reviews from me, including full review text
- All app purchases, including free apps, downloaded to the device
- All IP addresses assigned during downloads
- All books downloaded through Apple Books
- The Apple gift card number and amount applied to the account in 2015
- The dates/times and IP addresses of numerous app updates
- Hundreds of IP addresses used during my connections
- An export of all entries from Apple Calendar
- Documents remaining in iCloud
- Recent contacts within FaceTime
- Auto-stored contacts from Apple Mail
- Recipient email addresses accessed within Apple Mail
- Dates and times of outgoing email
- **My real name extracted from outgoing email headers**

I was disgusted. I had assumed that Apple had no interest in my daily usage of their hardware. The other data downloads from newer Apple ID accounts were cleaner, as I had stopped using official Apple services by 2018. However, the damage was done. Apple could easily uncover my true name behind my alias account, and share that with anyone they see fit.

What can be done? Apple stores all of this data about anyone who uses their services and equipment. They store it forever unless you request removal and termination of your account (which we will do together in a moment). I believe we can isolate ourselves from these risks while still taking advantage of macOS devices, as I will explain throughout this guide.

If you listened to my podcast, you already know that I previously moved away from Apple iOS mobile devices and adopted a secure GrapheneOS Android phone. I no longer use an iPhone, and I encourage my clients (and you) to do the same. You may also already know that I rely on a Linux laptop for daily personal usage. This works great for common tasks such as email, browsing, and secure communications. However, I still own a MacBook Pro with macOS. I refer to it as my production machine, and rely on it for Microsoft Office, Adobe programs, and other applications which do not work well on Linux. I also believe newer macOS devices with M1 or M2 processors have superior virtual machine functionality, as explained later. My daily OSINT investigations machine is a MacBook Pro.

Apple makes beautiful devices which perform very well. Their operating system is polished and fluid. Everything just works, and works well. Most of my clients are familiar with macOS and insist to stay within that ecosystem. I have no objection to that from a security stand, as I believe macOS may be the most secure operating system in the world. However, the privacy of their products is awful. Let's fix that.

This book will help you create a machine which does not send sensitive data to Apple. We will stop them from archiving our activities on the hardware which we have purchased. An Apple account will not be required in order to download applications and have full-functionality of the device. A name and physical address will never be associated with the device or Apple services. We will all take our privacy back, while possessing the perfect machine for our unique needs.

This entire book is designed for the reader interested in extreme privacy. 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 macOS 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 macOS 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 presents content which no one could dispute, then there is no need for this text.** Please read with an open mind and willingness to try new things. Let's begin.

CHAPTER ONE
ACCOUNT SANITIZATION

If you have purchased a brand new macOS machine, and have never used an Apple product in the past, you can skip this brief chapter. The goal here is to clean up any data we have given Apple in the past before we proceed to do things right. First, we should all request our data from Apple in order to understand the exposure.

- Navigate to https://privacy.apple.com.
- Sign in with your Apple ID and password.
- Select "Request a copy of your data" then "Get Started".
- Select all options and click "Continue".
- If prompted, choose a maximum file size of "1 GB".
- Click "Complete Request".
- Confirm any verification emails or text messages received.

In less than fourteen days, you should receive an email notification confirming your data is ready for download. Follow the included instructions to sign into your Apple account again and download the data. The file or files you receive should be compressed zip files. Double-clicking them should allow you to extract the contents. Every download will be unique for that user, but you should be able to navigate through the folders and access the files with a comma-separated value (CSV) extension. You should be able to open any of these files within the stock TextEdit application by right-clicking a file and choosing "Open With".

Peruse these files to see what details Apple has been storing about you. Save the files in a secure location for later analysis. If you want to eliminate this information from Apple's servers, conduct the following. Note that this is only possible if you plan to stop using this Apple ID. If you require this Apple ID for a mobile iOS device, you should not delete the account. If you only use this Apple ID for the computer which you plan to reset, conduct the following.

- Create a backup of any desired data within iCloud.
- Manually delete all possible files within your online iCloud account.
- Sign out of this Apple ID from within any devices which have access.
- Navigate to https://privacy.apple.com and sign in with your Apple ID.
- Select "Request to delete your account".
- Document the optional cancellation access code provided.
- Confirm any verification emails or text messages received.

A few weeks after you receive confirmation of account deletion, Apple should purge account details from their servers. We have no way to confirm this, so we must blindly accept their promise to do so. This was a short yet important chapter. Please take the time to complete this process, but you can proceed through the book while waiting.

CHAPTER TWO
HARDWARE CONFIGURATION

Once you have analyzed and deleted your existing data with Apple, it is time to discuss the hardware for your new private and secure macOS device. In a perfect world, you have an unlimited budget and are ready to purchase new hardware which has no association from your true identity to Apple. However, we do not live in that perfect world. Whether you are ready to purchase new equipment or need to recycle current hardware for future use, this chapter will explain all options and considerations. Let's start with new gear.

When Apple computers switched to their own ARM-based processors, instead of using trusted Intel chips, I was bummed. Numerous applications no longer functioned correctly and virtual machines were troublesome. Those days are over. The latest machines which include Apple's M-series processors are blazingly fast with low power consumption and minimal heat. Apps work better than ever. I have yet to hear my internal fans on my MacBook Pro, which was a daily occurrence on older machines. I now recommend the latest hardware available and believe the products with Apple's chips are superior to those with Intel processors. If you are buying new gear, make sure you are taking advantage of these benefits.

Selecting a machine is a very personal choice. Laptop options include the MacBook, MacBook Air, and MacBook Pro while desktops include the Mac mini, Mac Pro, and Mac Studio. I have never owned a Mac desktop, but I have purchased my share of Apple laptops. Today's least expensive small laptops will probably meet the needs for casual users, but the MacBook Pro models are all I will consider. I believe the latest 14" MacBook Pro laptops hit a sweet spot with productivity and value.

As I write this, the latest MacBook Pro's possess the M3, M3 Pro, and M3 Max processors. The previous generations possess the M1/M2 M1/M2 Pro, and M1/M2 Max chips. I am writing this from a 2021 14" MacBook Pro with the M1 Pro processor. What should you choose? Well, there are many conflicting opinions on this, and mine might not match yours. However, here is my advice.

If you will not be processing and exporting large 4K video files every day, or running four virtual machines simultaneously for several hours, then any M series device should be more than sufficient for your needs. If you want the latest machine for longevity of the hardware with operating system support, then the M3 series might be best for you. Either way, I recommend the Pro processors for most people. These provide more cores than the standard chips and more overall power. However, the Max processors would be overkill for most readers.

I believe most readers would get by with the minimal number of processor cores available with the latest 14" MacBook Pro, which is currently 11 (CPU) and 14 (GPU). Increasing the number of cores can assist with resource-intensive tasks, but most users would never take advantage of the power. You know if you need the extra boost.

The minimum option of 18 GB of RAM is also probably sufficient for most casual users. However, my machine possesses 32 GB of RAM because I work within multiple virtual machines simultaneously and parse large data sets daily. The standard 512 GB of storage should work fine for most, and external drives are much more affordable than embedded storage upgrades. My machine possesses a 4 TB internal storage drive because I work with large data sets (breach data) and need the fastest possible drive when working with the files. While these are great specs, I overpaid for the luxury.

I encourage everyone to possess a full backup drive, which is the size of your internal storage or greater. We will use this during the maintenance chapter, but it can also be beneficial for extra storage space. If you possess 512 GB of internal storage and a 1 TB external solid-state drive (SSD), you have the ability to clone your entire machine's data as a backup to the external drive, and extend your overall data limitations. I will explain more about this later, but I prefer the SanDisk Extreme line of external SSDs.

I firmly believe that the newer M2/M3 2023 models are worth the minimal current price increase from the previous 2021 generation. However, I have seen brand-new previous-generation laptops deeply discounted at various Apple resellers. If I were buying a new machine today, it would definitely be the newer M-series model. If you already possess an M1 machine which meets your needs, I see no reason to upgrade to the M2 or M3. The real-world comparisons will be negligible. If you upgrade from an Intel processor to an M1/M2/M3, I believe you will be shocked at the difference. My point is that I recommend a machine with newer ARM-based M1, M2, or M3 processors for most readers. While you can use older hardware with Intel chips, and almost all of this guide will still apply, you are missing out on a phenomenal increase in power and battery life.

I focus on MacBook Pro laptops because they are the most common request I receive from my clients. However, everything presented within this guide would also apply to any modern macOS device. This includes the MacBook, MacBook Air, Mac mini, Mac Pro, Mac Studio, and any other system which supports macOS. This guide does not apply to iOS devices such as the iPhone and iPad.

We should now have the new vs. used conversation. If you buy a new machine properly, and apply the methods explained throughout this book, Apple will assign no history of its usage to you. When I discuss mobile devices, I only consider new devices which have never been used by anyone else. This is due to embedded unique identifiers which are constantly shared with cellular companies. Purchasing a used mobile device from a criminal being monitored by the government could make you a target. The same could be said about a macOS computer, since it possesses a unique serial number which is constantly shared with Apple. However, there are major differences.

You cannot prevent Apple from knowing the serial number of an iOS device, and Apple requires an active Apple ID connection in order to download applications. That device is constantly sending unique identifier information to numerous parties. A macOS computer also sends out the serial number by default, but we can mostly block that if desired. Also, our finished macOS device will not require an Apple ID, which will minimize sensitive data storage about your usage.

Back in our perfect world, it will always be better to purchase new equipment and start fresh. However, reformatting the drive of an existing system, and applying better privacy hygiene is much better than doing nothing at all. If your only option is to re-use existing equipment which previously possessed an Apple ID, I do not think that is the end of the world. We will clean it up and prevent further abuses together. It all comes down to your desire for extreme privacy, level of paranoia, and overall goals. Don't let a guide titled "Extreme Privacy" prevent you from taking the steps which are available to you, even if not optimal for those under aggressive threats. **Whether you have elected to purchase a new computer or wish to continue with existing hardware, the rest of this chapter still applies to you.**

Next, we should discuss purchase options. I would never buy an Apple device from their website. Their fraud detection algorithms will force you to use a credit card in a true name, and shipment to a CMRA or PO Box will flag the purchase for review. Apple will keep a detailed log of the purchaser's name, physical address, IP address and computer characteristics forever, and associate all of it with the serial number of the unit. Fortunately, we have better options.

I almost always purchase Apple devices for myself and my clients with cash inside an official Apple store. You will receive skepticism and judgement when you pull out $2,000 in cash, but I don't mind that. Purchasing with cash is anonymous. If they force a name for the receipt, give them whatever you want. The device is under warranty based on the sales date and serial number, regardless of the owner. If I must order a device online, I prefer B&H (bhphotovideo.com). They are an official Apple reseller and often offer discounts on devices. The ordering characteristics and shipping addresses are much less scrutinized by them than Apple. When B&H flags a purchase for review, they call you and ask a few questions to confirm the order. Apple simply deletes it and offers no recourse. I have successfully ordered numerous MacBook Pro's for clients from B&H while using secondary credit card names and random CMRA's.

I will now assume that you possess your desired macOS device. Regardless of its condition or previous usage, I believe every reader should now reformat the drive and apply a fresh install of the operating system. This ensures that we are all on the same page for an identical experience. Make sure you have completely backed up all important data before continuing, as the following processes will erase everything on the drive. If you do not have a proper backup strategy, you may want to read the backup section of the Updates & Maintenance chapter before proceeding. I will now assume you have a backup of all important data.

Before proceeding, I want to address an additional layer of security for readers who possess on older Intel-based machine. While newer M-based devices already possess firmware which is set to the optimal settings, older machines do not have secured firmware by default. Locking the firmware will require a password to be entered in order to access the firmware menu during future access attempts, and will restrict the device to booting only from the specified internal disk. This can minimize damage from physical attacks which attempt to access data in a forensic fashion. The following steps should only be applied to Intel-based machines, such as devices made prior to

2020. Note that some machines may require internet access via Wi-Fi or ethernet cable in order to complete these steps.

- Turn the device completely off.
- Hold "Command" and "R" simultaneously until the device boots.
- Select the user account and enter password if required.
- Click "Utilities" in the menu bar and select "Startup Security Utility".
- Click "Turn On Firmware Password".
- Enter a strong password then click "Set Password".
- Document this password within your password manager.
- If present, ensure "Secure Boot" is set to "Full Security".
- If present, ensure "Allowed Boot Media" is set to "Disallow...".
- Close the window, then click the Apple menu and choose "Restart".

Finally, we can now wipe out our machines. If you purchased a brand new M1, M2, or later device which does not possess an existing Apple ID account, and has never been turned on, you can skip to the next chapter. If you are working from an existing device, regardless of the processor type, we should consider several tasks. First, update the operating system to the latest available version. As I write this, my machine possesses Apple's Sonoma version of macOS, specifically 14.0. By the time you read this, that exact number will change. I always recommend the latest stable version available, and avoid any beta (test) builds. **The following assumes you possess macOS Sonoma as your operating system and are able to update to the latest version of macOS Sonoma.** This will require devices made after 2018, but unsupported devices can still take advantage of most of this book using previous versions of macOS. You will need to slightly modify the steps for your specific operating system.

Open the "System Settings" application; click the "General" option and then the "Software Update" setting. Allow your machine to download and install all available updates, and then reboot. Once your machine is fully updated, conduct the following within macOS to reinstall a fresh version of the operating system.

- Open the "System Settings" application.
- Click the "General" option and then "Transfer or Reset".
- Click the "Erase All Content and Settings" button.
- Enter your password within the "Erase Assistant".
- Confirm all warnings and allow the process to complete.

Previous versions, such as Ventura, Monterey, or Big Sur, should also provide the option to reset the system through either the "Erase Assistant" or "Recovery" mode. You will need to research options for your non-Sonoma version. Upon reboot, you should possess a clean installation of macOS ready for initial configuration. Do not take any actions yet, as there are many things to tweak right from the initial welcome menu, as explained in the next chapter.

CHAPTER THREE
OS CONFIGURATION

I will now assume that you either have a brand-new computer or a recently-reset device. Either way, it should appear as a new installation when turned on for the first time. Regardless of your processor type or history with the machine, the following applies as if you were a new user.

I recommend that users do not connect internet to the new system until after a firewall is installed, as explained in the next chapter. This includes any Wi-Fi or ethernet connection. If building a virtual machine (VM), as explained later, I would disable internet connectivity to it until the firewall is installed within the VM, but it is not as vital as doing so from a host machine. I offer much more on the concerns with this in the next chapter. I will now assume that you are ready to launch your new macOS installation for the first time, without any internet connection.

Upon launching macOS for the first time, your experience may be unique from mine. Updates to the operating system from Apple and specific hardware configurations could present minor variations from the steps outlined here. I took the following actions within a new macOS Sonoma installation, which had not been updated to the latest release. It was the original stock Sonoma version 14.0. If Apple prompts you to connect to nearby Wi-Fi, simply cancel the request and continue through the steps.

- Select desired language and click the right arrow.
- Select country and click "Continue".
- Click "Customized Settings".
- Confirm preferred language.
- Confirm location.
- Confirm dictation (required).
- Click "Not Now" for Accessibility options.
- If prompted, choose "My computer does not connect to the internet".
- Click "Continue" and "Continue" again if requested to connect to the internet.
- Click "Continue" for Data & Privacy notification.
- Click "Not Now" for the Migration Assistant.
- Click "Set Up Later" to bypass the Apple ID requirement.
- Confirm by clicking "Skip".
- Click "Agree" to the Terms and Conditions.
- Confirm by clicking "Agree".
- Create a local computer account. This should be a generic name, such as "Laptop" or "Computer", and should include a very strong password which you can remember. I never provide any password hint to this screen. Click "Continue" when finished.
- Do not enable "Location Service" and click "Continue".

- Confirm choice by clicking "Don't Use".
- Select your desired time zone and click "Continue".
- Deselect all analytics options and click "Continue".
- Click "Set Up Later" to bypass "Screen Time" settings.
- Disable Siri and click "Continue".
- Choose your desired screen mode and click "Continue".

You should now see the macOS desktop which is ready for customization. Since you have no internet connectivity, there should be no notifications of pending updates. During the next chapter, we will set up our firewall to block invasive data gathering while still being able to update the operating system. For now, let's focus on numerous privacy and security tweaks we can make within the OS itself. The following steps configure Wi-Fi and Bluetooth.

- Launch "System Settings" from the Dock.
- Select "Wi-Fi" from the left menu and disable it.
- Disable both "Ask to join networks" and "Ask to join hotspots".
- Select Bluetooth from the left menu and disable it.

Next, I want to configure the operating system's firewall. This is much different than the firewall we will install in the next chapter. This is only responsible for the way the operating system treats incoming connections. The following steps enable the firewall and configure it to block incoming connections unless we specifically allow them when prompted. It also stops the OS from confirming incoming requests for information.

- Select "Network" from the left menu and select "Firewall".
- Enable the Firewall and click "Options".
- Disable "Automatically allow built-in software to receive...".
- Disable "Automatically allow downloaded signed software to receive...".
- Enable "Stealth mode".
- Click "OK".

Next, I like to disable all notifications possible. I do not want sensitive applications, which will be installed later, to display content on the screen when I am not around or when someone is over my shoulder.

- Select "Notifications" from the left menu.
- Change "Show previews" to "Never".
- Disable "Allow notifications when the device is sleeping".
- Disable "Allow notifications when the screen is locked".
- Disable "Allow notifications when mirroring or sharing the display".
- Open each application, disable notifications, and click the arrow to return.

I also prefer to disable any unnecessary sounds with the following steps.

- Select "Sound" from the left menu.
- Change "Alert volume" to the minimum setting.
- Disable "Play sound on startup".
- Disable "Play user interface sound effects".
- Disable "Play feedback when volume is changed".

The following should already be disabled by default, but let's make sure.

- Select "General" from the left menu.
- Select "AirDrop & Handoff".
- Disable "Allow Handoff between this Mac and your iCloud devices".
- Confirm AirDrop is set to "No One".
- Select "General" from the left menu.
- Select "Sharing".
- Confirm all options are disabled.
- Select "Siri & Spotlight" from the left menu.
- Confirm "Ask Siri" is disabled.

If you want to truly ensure that Siri is not listening in on your activity, you can conduct the following, which may be redundant.

- Select "Siri & Spotlight" from the left menu.
- Click "Siri Suggestions & Privacy".
- Click each option and disable all toggles, then click "Done".

The next section is completely optional, and may not be appropriate for everyone. Spotlight is an indexing and search service offered by macOS. It allows you to quickly search for document content and file names, which can be especially convenient for finding desired files. It can also be invasive. I do not like Apple searching through and indexing my documents. I do not want macOS to possess a database with my sensitive content. I do not know if they are sending any of that data to their servers when they attempt to collect other usage characteristics every minute while the device is on. While we will block this behavior within the next chapter, I prefer to disable the basic features of Spotlight with the following steps.

- Select "Siri & Spotlight" from the left menu.
- Disable all options within the Spotlight area.
- Click "Spotlight Privacy".
- Click the "+" in the lower-left.
- Change the dropdown field to "Macintosh HD".
- Click "Choose", confirm with "OK", and click "Done".

If you take these steps, your computer will no longer properly search through your files. I see this as a benefit, but you may find it to be a hinderance. Make this choice carefully, but also know that you can always reverse these steps. Since I know I do not want Apple's Spotlight service running, I also conduct the following to lock in these settings.

- Open "Finder" from the Dock.
- Select "Applications" from the left menu.
- Double-click the "Utilities" folder.
- Double-click "Terminal" to open the program.
- Enter "`sudo mdutil -i off /`" (without quotes) and press return.
- Enter "`sudo mdutil -E /`" (without quotes) and press return.

If you ever want to reverse this, conduct the following.

- Enter "`sudo mdutil -i on /`" (without quotes) and press return.
- Enter "`sudo mdutil -E /`" (without quotes) and press return.

If you disable Spotlight completely, your device will stop analyzing and ingesting every change you make to your files. It will also stop adding data to their hidden index of your most sensitive content. In a future chapter, I will present my custom search script which can be used in absence of Spotlight.

The next consideration is Apple's Gatekeeper service. This feature sends data about the applications present on your system to Apple. Whenever you open an application for the first time, or after it has been updated, macOS determines whether or not Apple has verified the software. Stock Apple applications always launch without issue, but Gatekeeper prevents all unknown apps from opening. Even applications from identified developers can produce warnings when first launched, depending on your security settings, and you may need to authorize their use. If you download a legitimate application which has not been blessed by Apple, your machine will initially refuse to open it. Right-clicking the program will typically bypass this, but I still find it annoying.

My larger concern is that I do not want Apple keeping track of every application on my machine, and I desire to block the constant connections announcing my software habits to Apple's servers. Most of our computers appear very unique based on this fingerprint, which would make it trivial to track us, even without an Apple ID account.

Therefore, I choose to completely disable Gatekeeper. Some may see this as a security risk, and I respect that opinion. If my older relatives adopted a macOS system, I would not want this setting disabled on their machines. It could save them from executing a malicious program. Since I do not install questionable applications or download software from shady sources, I believe my risk is minimal. Also, I commonly execute trusted open-source applications which are not Apple-approved, and I enjoy the lack

of roadblocks when I want to use the programs. Consider your own risks before proceeding.
The following command within Terminal will disable Gatekeeper.

```
sudo spctl --master-disable
```

The following command re-enables Gatekeeper.

```
sudo spctl --master-enable
```

The following command displays the current status of Gatekeeper.

```
spctl --status
```

Once Gatekeeper is disabled, you should see the following options within "System Settings" > "Privacy & Security".

Since I do not have an Apple ID on my machine, I cannot use the App Store, so I do not want that option selected. Since I have disabled Gatekeeper, I can now choose the option to allow applications from "Anywhere".

Let's conduct a few more configurations within System Settings.

- Select "Privacy & Security" from the left menu.
- Select "Analytics & Improvements" and verify all are disabled.
- Select "Privacy & Security" from the left menu.
- Select "Apple Advertising" and disable "Personalized Ads".
- Select "General" from the left menu.
- Select "Software Update".
- Click the "i" in the circle and deselect everything.

The last setting stops macOS from constantly checking for updates until we are ready to install them. The following forces your operating system to use the Network Time Protocol Project's time synchronization server instead of Apple's network.

- Select "General" and choose "Date & Time".
- Click "Set..." next to "Source" and enter your password if prompted.

- Change the time server to "pool.ntp.org" and click "Done".

Next is likely the most important setting within this chapter. By default, the data stored on your macOS system is not encrypted. Physical access to your computer using sophisticated forensic equipment could extract your data. If you lose your laptop, or it is stolen, there is a chance that the culprit could acquire your sensitive documents. The best way to prevent this is to apply full-disk encryption through Apple's FileVault with the following steps.

- Select "Privacy & Security" from the left menu.
- Click "Turn On..." next to "FileVault".
- Enter your system password and click "Unlock".
- Choose "Create a recovery key and do not use my iCloud account".
- Document this recovery key somewhere safe and click "Continue".

Your device will now encrypt the drive, including all data stored within it. This is a vital piece of protection which I believe should be enabled by default.

You may have noticed an option called "Lockdown Mode" within this screen. It offers an additional layer of protection from cyber-attacks. On the surface, this may seem like a desired feature. However, I do not use it. Most of the benefits of this setting apply to Apple's infrastructure including FaceTime, iMessage, Photos, and other macOS-provided applications. I do not use any of these and neither should you. Therefore, most of the protection would not apply to us. Furthermore, the setting restricts our use of some external devices. Therefore, I do not recommend it.

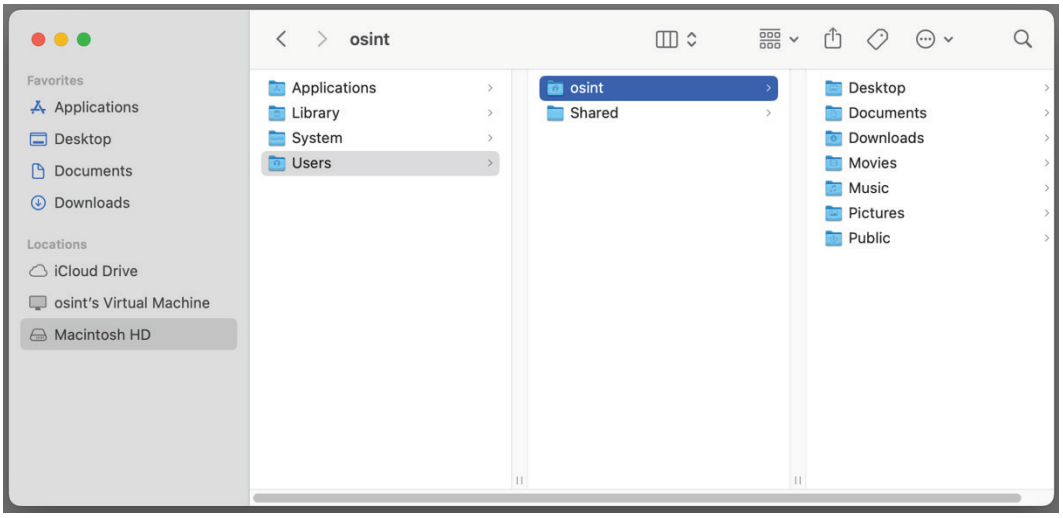
Your macOS device should now be more private and secure than it was, but I have a few additional settings I like to apply. These are all personal preferences, and you may want to tweak these differently.

- Select "Desktop & Dock" from the left menu.
- Disable "Show suggested and recent apps in Dock".
- Disable "Show recent apps in Stage Manager".
- Select "Wallpaper" from the left menu.
- Choose a solid color instead of the default macOS image.
- Select "Lock Screen" from the left menu.
- Change "Start Screen Saver when inactive" to "Never".
- Change "Turn display off on battery when inactive" to "For 1 hour".
- Change "Turn display off on power adapter when inactive" to "For 1 hour".
- Change "Require password after..." to "Immediately".

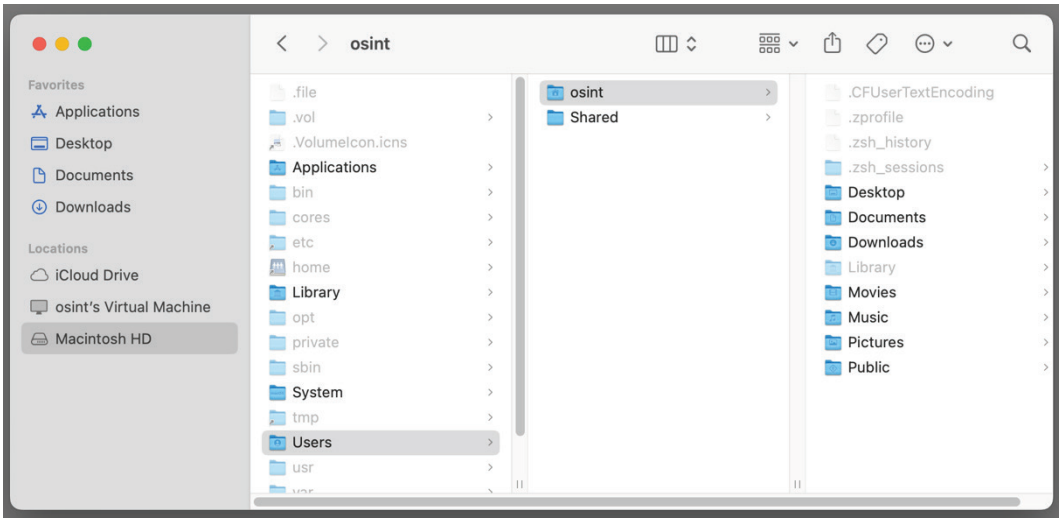
These settings prevent the macOS screen saver from kicking in, and instead disable the display after a set amount of time. This also makes sure that your password is required the moment a screen is disabled or the device is placed into standby mode, such as closing the lid of a laptop.

Next, I want to modify the default way in which Apple allows you to see the data stored within your device. Apple is proud of the "simple" features of macOS. Things just work and you are not bombarded with complex options. However, you are also severely restricted. As one example, macOS hides all "hidden files" from view within Finder. While most users do not care about this data, I do. Much of my most important data is within a hidden "Library" folder to which I have no access. Let's fix that.

- Open Finder and select the "Macintosh HD" in the left menu.
- Select "Users", and then your device's username.
- Notice the view of this folder, which should appear similar to the following.

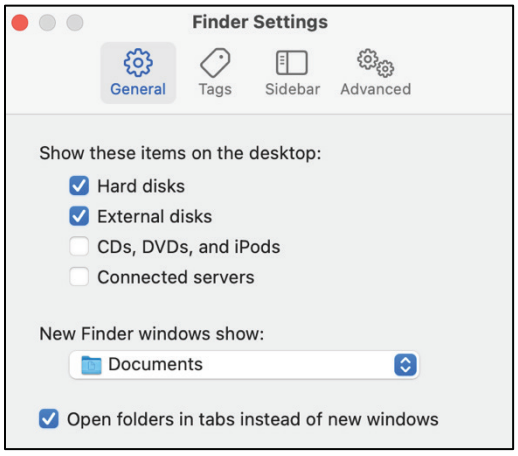


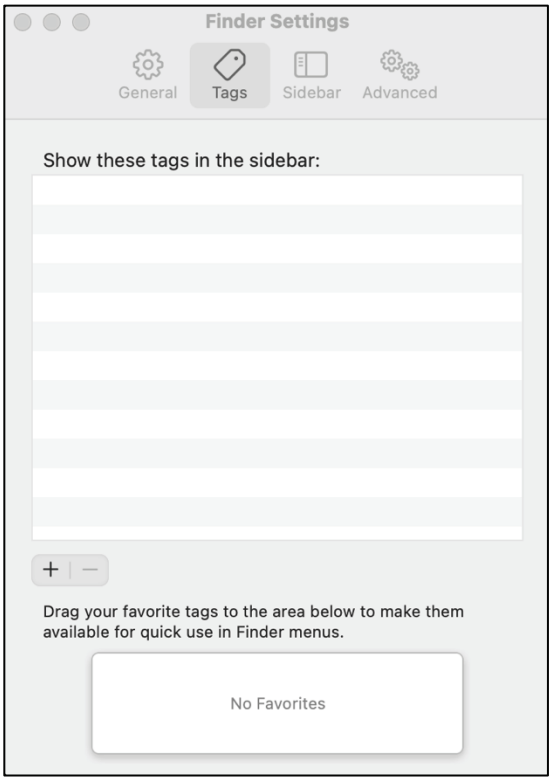
On your keyboard, press and hold shift + command + . (period). Your view should change similar to the following image. You can now see all files determined to be hidden by macOS. This will be vital once we backup our system.

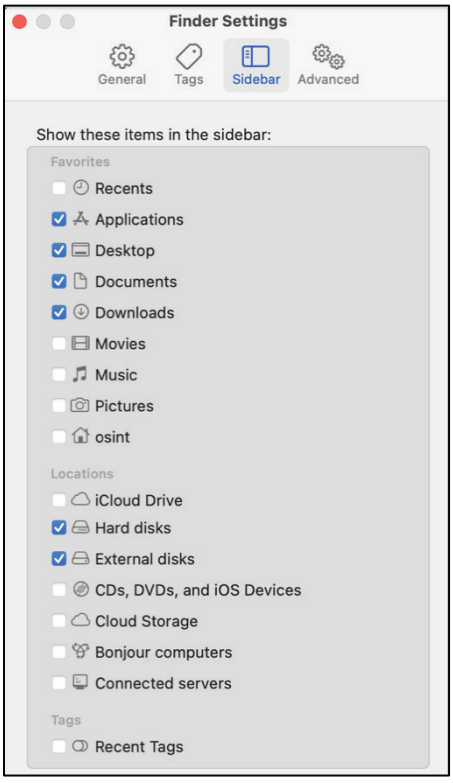


While we are in Finder, let's modify some settings.

- Click Finder in the upper-left menu bar and then select "Settings".
- Choose the "General" tab and consider my choices presented below.
- Choose the "Tags" tab and consider my choices presented below.
- Choose the "Sidebar" tab and consider my choices presented below.
- Choose the "Advanced" tab and consider my choices presented below.







Finally, I want to modify the Dock and its contents. By default, macOS presents a Dock with large icons at the bottom of the desktop. They also conveniently promote their own applications while ignoring more valuable shortcuts such as Terminal. I keep a few commands ready which will make my desired changes. First, I prefer my Dock on the left side of my screen, similar to my Linux machine. The following sets the Dock for left alignment, and then refreshes the Dock settings.

```
defaults write com.apple.dock orientation left; killall Dock
```

Next, I like to remove all of the undesired stock applications from the Dock and replace them with the programs which I use most often. The following Terminal command removes all icons and only adds Safari, Terminal, and System Settings.

```
defaults write com.apple.dock persistent-apps -array; defaults
write com.apple.dock persistent-apps -array-add
'<dict><key>tile-data</key><dict><key>file-
data</key><dict><key>_CFURLString</key><string>/Applications/S
afari.app</string><key>_CFURLStringType</key><integer>0</integ
er></dict></dict></dict>'
defaults write com.apple.dock persistent-apps -array-add
'<dict><key>tile-data</key><dict><key>file-
data</key><dict><key>_CFURLString</key><string>/System/Applica
tions/Utilities/Terminal.app</string><key>_CFURLStringType</ke
y><integer>0</integer></dict></dict></dict>'
defaults write com.apple.dock persistent-apps -array-add
'<dict><key>tile-data</key><dict><key>file-
data</key><dict><key>_CFURLString</key><string>/System/Applica
tions/System
Settings.app</string><key>_CFURLStringType</key><integer>0</in
teger></dict></dict></dict>'; killall Dock
```

This presents extremely large icons, which I do not like. The following command decreases them to a size of "40" instead of the default "128".

```
defaults write com.apple.dock tilesize -integer 40; killall Dock
```

While this command replaced the icons before the Dock separator, it did not make any changes to the Downloads folder and Trash options at the bottom. I prefer a shortcut to the Applications menu which can be seen as a list. The following applies this setting.

```
defaults write com.apple.dock persistent-others -array-add
"<dict><key>tile-data</key><dict><key>file-
data</key><dict><key>_CFURLString</key>
<string>file:///Applications/</string><key>_CFURLStringType</k
ey> <integer>15</integer></dict><key>file-
type</key><integer>3</integer>
<key>showas</key><integer>3</integer></dict><key>tile-
type</key><string>directory-tile</string></dict>"; killall
Dock
```

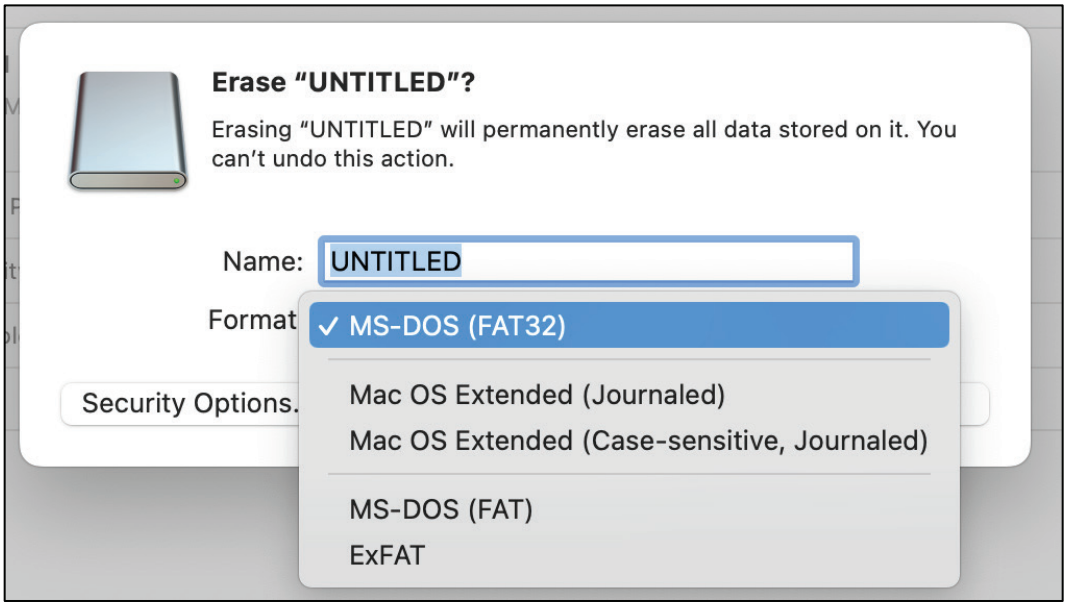
If you prefer this new icon to only look like a folder, and not the first Application icon within the Applications, right-click this new icon and select "Folder" under "Display as". You can now add your desired applications to the Dock as we work through the book by simply dragging and dropping icons from the Applications folder within Finder to the Dock itself, in whichever order you like. If you do not like the way this all looks, the following command returns everything back to the default settings.

```
defaults delete com.apple.dock; killall Dock
```

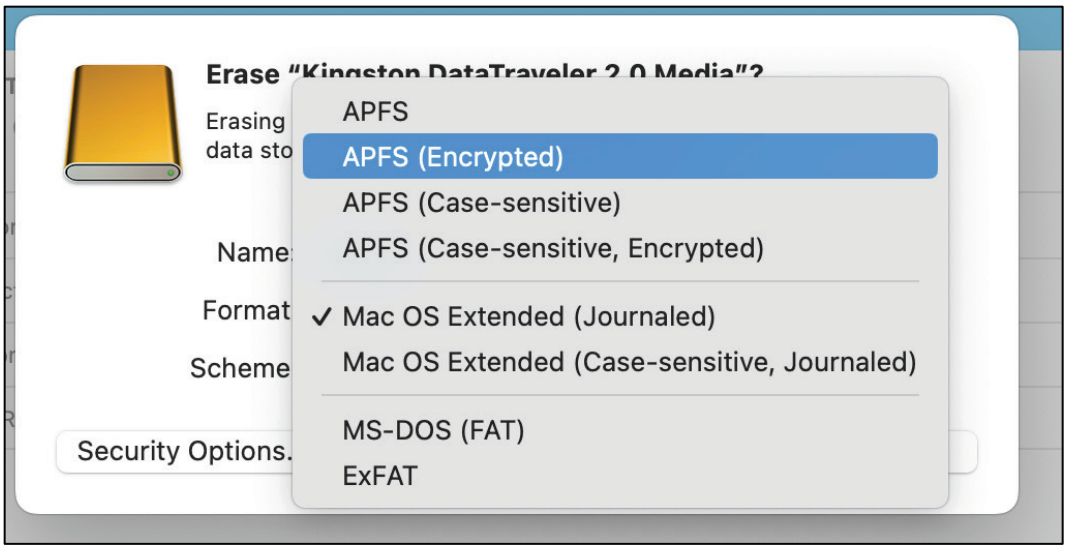
You should now have a stable and protected operating system ready for third-party applications. You still have no internet connectivity, but all of the privacy and security basics are in place. As a reminder, you have not associated your new machine with an Apple ID, and hopefully you never will do so. Without this connection, Apple has limited capabilities associating the activity occurring on your machine with a specific user or account. By refusing to attach an Apple ID, you are much more private and secure. I have not assigned an Apple ID to my current (or previous) machine, and I never will again in the future.

Encrypted External Drives

In a later chapter, I explain how I use an external USB SSD drive for backups, and the importance of encrypting the data within this drive. I believe this applies to any external drive which we connect to our macOS devices. USB drives are often lost or stolen. Therefore, we should always encrypt the data stored on them. This is not always easy. Sometimes, macOS hides the settings we need to protect an external device. As an example, I inserted a small USB drive which was formatted as "FAT32", which is common for universal drive access. I wanted to erase the drive and encrypt it. However, the Disk Utility application (Applications > Utilities) only displayed the following options. Right-clicking the drive in Finder also did not present an option to encrypt the drive.



The first step to take within the Disk Utility application is to select "Show All Devices" under the "View" menu. Next, select the device (not the formatted volume) within the left menu and click the "Erase" button. This may still only present volume formats which cannot be encrypted. Be sure to change the "Scheme" to "GUID Partition Map". You should now see and option of "APFS (Encrypted)" under "Format". This option will encrypt the entire external drive with macOS encryption. I believe this is the best option for users who will only need to access this drive from a macOS system.

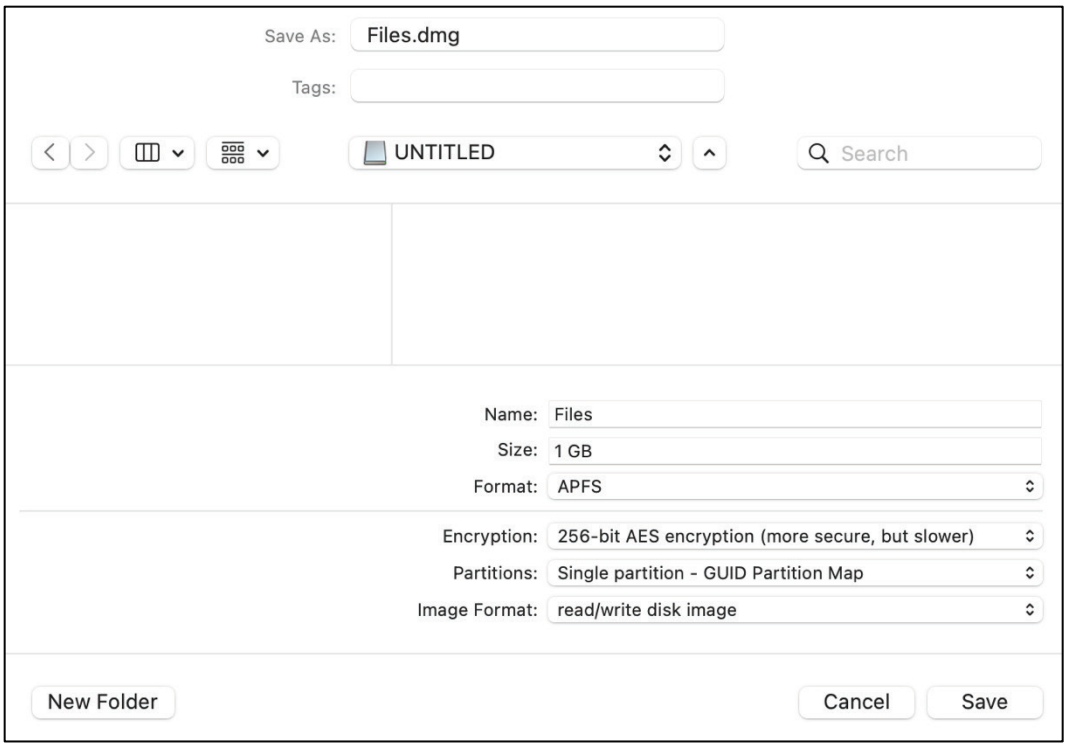


Encrypted Containers

You may have an external drive which needs to be readable within any operating system such as macOS, Windows, or Linux. It may have a universal format such as FAT32 or exFAT. You may keep data on this drive which is not sensitive and does not need to be encrypted. However, you may later want to protect a small portion of the drive which can only be read within your macOS system. As an example, I inserted a USB drive formatted with FAT32. I returned to the Disk Utility application and selected the volume on this drive, which was listed as "UNTITLED". I selected "File" > "New Image" > "Blank Image" from within the file menu. I then conducted the following.

- Choose the location of the external drive.
- Provide a name for the file, such as "Files.dmg".
- Provide a name for the container, such as "Files".
- Choose the maximum size for the container, such as "1000 MB".
- Choose a format of "APFS".
- Choose an encryption level of "256-bit Aes…".
- Enter and verify the password you want for the container and click "Choose".
- Choose a "Single " partition.
- Choose an Image Format of "read/write disk image".
- Click Save".

This will generate a new 1000 MB container file within the external drive. It will be encrypted and the password will be required any time you want to open it (by double-clicking the file). The following image displays these options during creation.



The drive now possesses a file called "Files.dmg". Double-clicking this file prompts me to enter the encryption password. Entering the correct password mounts a container called "Files" which I can access from within Finder. Anything I store within this volume (container) will be protected even though my external drive is not encrypted. The size limit of the container in this example is only 1000 MB. Always consider your future storage needs.

Many readers may question my reliance on Apple's proprietary encryption over universal open-source options such as VeraCrypt. I believe VeraCrypt is an amazing program, and I use it within my Linux systems. However, I typically do not recommend it for macOS users. Apple's encryption options work much better and are more efficient on Apple hardware. VeraCrypt for macOS requires a third-party software installation called OSXFUSE. I have witnessed many issues when attempting full-disk encryption within VeraCrypt for macOS, especially newer machines. If you will always have access to a macOS device, I believe Apple's default encryption options are superior to VeraCrypt for macOS. To be clear, I still use VeraCrypt for other operating systems.

For most external devices, I recommend full-disk encryption within macOS, as previously explained. For those readers who know they need a universally-formatted device, then the container option exists for you. Next, we tackle the most important chapter of this guide.

CHAPTER FOUR
FIREWALLS

I briefly mentioned the embedded macOS firewall in the previous chapter. Its purpose is to block unauthorized INCOMING connections to your system. However, that is only half of the story. A much more important issue is the OUTGOING connections from your operating system and applications. The moment you enable an internet connection to your macOS device, the system begins sending information to Apple's servers. This could include unharmful data such as a check for updates, or sensitive information from emails, contacts, and search history. Third-party applications are just as bad. Many of the apps we trust are sending telemetry and usage information about us behind our backs. My goal is to present options which reduce or eliminate the exposure. The way we can do this is with third-party firewall applications.

Software firewalls permit us to "Allow" or "Deny" any outgoing connection from applications or the operating system. We can also use them to temporarily block transmissions to see if anything breaks, or provide permanent blocks for things we know we never need. This should make more sense when we configure each option.

In previous writings, I relied exclusively on a paid program called Little Snitch. While I still prefer this option, and use it every day, I want to expand my tutorials to a free application called Lulu. You should only use one of these options, and never both. Please read through this entire chapter before you decide which path you will take. After presenting the manual approach to firewall configuration, I will offer pre-built settings which can be imported into my preferred program.

Up until now, you have not enabled any internet connectivity on your fresh macOS installation. You need to install your chosen firewall application which will require internet access in order to download the installation file. However, connecting the internet to our new machine exposes us to invasive data connections which would have otherwise been prevented from the firewall application which you are trying to install. This is quite a Catch-22.

My preference is to always download the necessary files from another machine and transfer them to a USB drive for easy installation. If you do not have access to another machine, it would not be the end of the world to connect your new device to the internet for this purpose. However, this is Extreme Privacy, so I will assume that you will find a way to download the desired application on another device and transfer it over via USB. This prevents any undesired network connections which could make your machine feel "dirty". The following links will obtain the necessary files.

Little Snitch: https://www.obdev.at/products/littlesnitch/download.html
Lulu: https://objective-see.org/products/lulu.html

Please remember that you only want ONE of these options installed at any time. Since I prefer Little Snitch, I will start with it.

Little Snitch 2024 Landscape

Before I begin with Little Snitch, I want to address some changes since the original publication of this guide. In May of 2024, Little Snitch version 6 was released. Many readers of this guide likely possess version 5. Since version 6 is a paid upgrade, many readers are asking about the advantages of upgrading, if any. Furthermore, a free version called Little Snitch Mini has been released, and is quite robust for being free. Let's dissect each consideration.

Little Snitch 5 still works great and provides all of the advanced firewall protection which we need. It will work fine for you if you do not want to upgrade. You just will not receive the new features within version 6 (which you may not use anyway). Little Snitch does offer a discounted upgrade if you recently purchased the older version. While I upgraded in order to be on the latest version for testing, I do not believe it is needed for all readers. If purchasing the app for the first time today, you will automatically receive the latest version (6).

Little Snitch 6 is very similar to version 5 with two exceptions. First, a facelift and slight graphical modification makes the application smoother and slightly easier to navigate. However, I do not really care about that. The new feature which allows encrypted DNS queries is a substantial upgrade for SOME readers, but not all. If you followed my guide about firewalls and implemented a home pfSense firewall with encrypted DNS, you do not need this new feature. If you implement the custom DNS configurations within this guide, you also do not get any real benefit from this new option. In fact, it will override your hard work and you will no longer receive the benefits of your custom NextDNS filtering. Also, this setting will have no impact on your browser traffic if you rely on Firefox's strict DNS settings explained later in this guide. If you have never changed your DNS, ignored the advice about DNS in this guide, and rely only on your ISP to provide DNS, then this is a great feature within Little Snitch. However, I doubt many of my readers need this feature because I believe we have better options, as explained in this guide. **I do not enable Little Snitch's DNS encryption on my machine.** I follow the DNS protocols explained later.

Little Snitch Mini is a free minimalistic version of the full application. It simplifies the process for blocking outgoing connections from applications without much of the confusing dialogues. However, it requires download through Apples App Store and all updates must be delivered through Apple. Since I do not recommend applying an Apple ID to any macOS device, I cannot recommend the free version of Little Snitch Mini. If a direct download option should surface, I will revisit this recommendation.

The custom Little Snitch configuration file presented in a moment works the same on version 5 or 6, but does not work with Mini.

Little Snitch Configuration

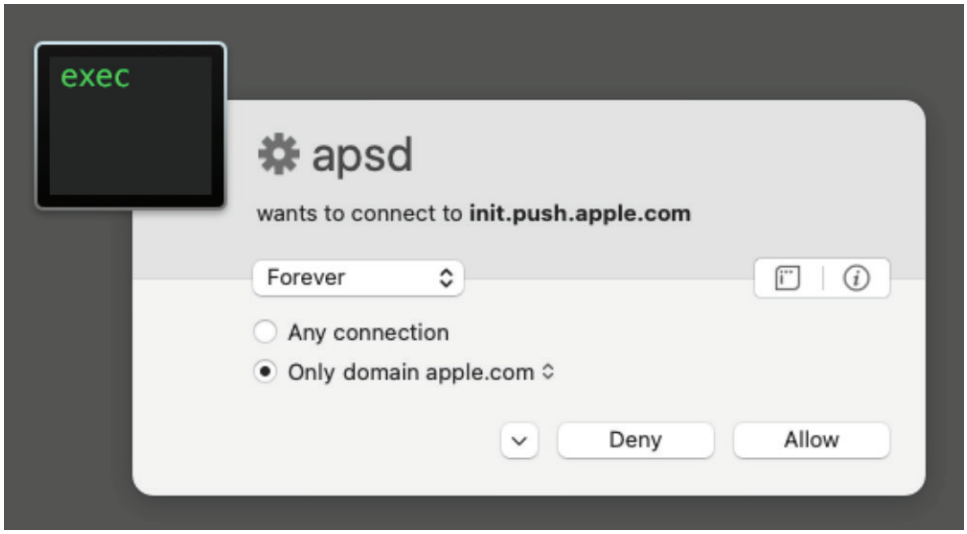
I have been using Little Snitch for many years. My usage began as a way to block Microsoft Office and Adobe from constantly sending out information about the documents which I was creating. As Apple became more invasive into our usage of their products, I now rely on Little Snitch to prevent Apple from getting my data along with third-party applications.

You can either follow along with me during the manual configuration of Little Snitch, or wait until the end when I provide a file which can be imported to replicate my setup right away. I always encourage people to understand and experience the manual process, but I also know how frustrating it can be when Little Snitch initiates dozens of confirmation screens because macOS is trying to suck up your data. I will simply provide my steps here, and you can decide which path you want to go.

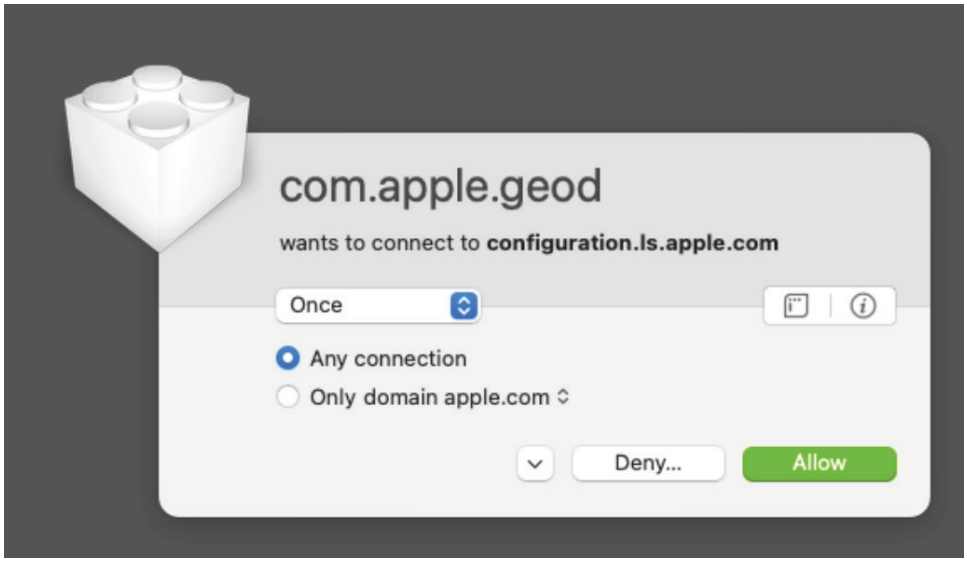
I navigated to the Little Snitch website and downloaded the latest version of the software to my USB drive from another machine, then transferred the file to my new build. Installation was straight-forward, but I did deviate from the default options. I took the following steps.

- Open Finder and navigate to the downloaded Little Snitch installation file.
- Double-click it and drag the Little Snitch icon into the Applications folder.
- Close the Little Snitch window.
- Navigate to the Applications folder with Finder.
- Double-click the Little Snitch application and confirm "Open".
- Accept the license agreement and click "Install".
- Click "Open System Settings" when prompted.
- Click "Allow" when prompted within System Settings.
- Enter the password to the system and click "OK".
- Click "Allow" for network content access.
- Close any notification windows.
- Click "Start Tour", then "Next" six times.
- Click "Continue" then select "Alert Mode", and click "Next".
- Disable both "macOS Services" and "iCloud Services" and click "Next".
- Click "Close" and then "Demo Mode" on the popup.

You are now running Little Snitch in demo mode. This mode is completely free and all features are available. However, it will stop functioning after three hours, or a reboot, unless a license is purchased. You should now see a window with a summary of all connections. If your internet connectivity is disabled, this window should be empty. I enabled internet connectivity via Wi-Fi and immediately received the following notification from Little Snitch.



This notified me that Apple is trying to send data out to its push service, likely to see if I have any pending FaceTime notifications. Even though I do not use FaceTime, and an Apple account is not registered to my device, Apple still wants to collect data and send it to its servers. I do not want Apple to ever connect to their push services, and I never plan on using FaceTime, so I selected "Forever" and "Any connection", then clicked "Deny". This immediately presented the following.



This notifies me that Apple is trying to send data out to its location service, likely to collect my IP address and Wi-Fi connection details for geo-location. Even though I disabled location services, Apple still wants to collect data about my location every hour and send it to its servers. I do not want Apple to ever connect to their location server, and I never plan on using their location services, so I changed "Once" to "Forever"; selected "Any connection", then clicked "Deny". This prompted me to ensure I wanted that setting, as Apple deems this to be a vital service. I chose "Deny Anyway" to confirm the setting.

Little Snitch then continued to alert me to the dozens of connections Apple was trying to create from my machine to its servers. This is why firewalls can be overwhelming and Little Snitch offers the "macOS Services" option. If we had selected it, the firewall would have allowed all of these "normal" connections. However, I do not want Apple sucking up information about me every hour of the day.

I blocked all requests through Little Snitch with the following exceptions.

mDNSResponder: "Allow" to "Forever" connect to "Any Connection"
mDNSResponder: "Allow" to "Forever" connect from "Local network"
timeD: Allow to connect to "pool.ntp.org"
Safari: "Allow" to "Forever" connect to "Any Connection"
Little Snitch: "Allow" to "Forever" connect to "obdev.at"

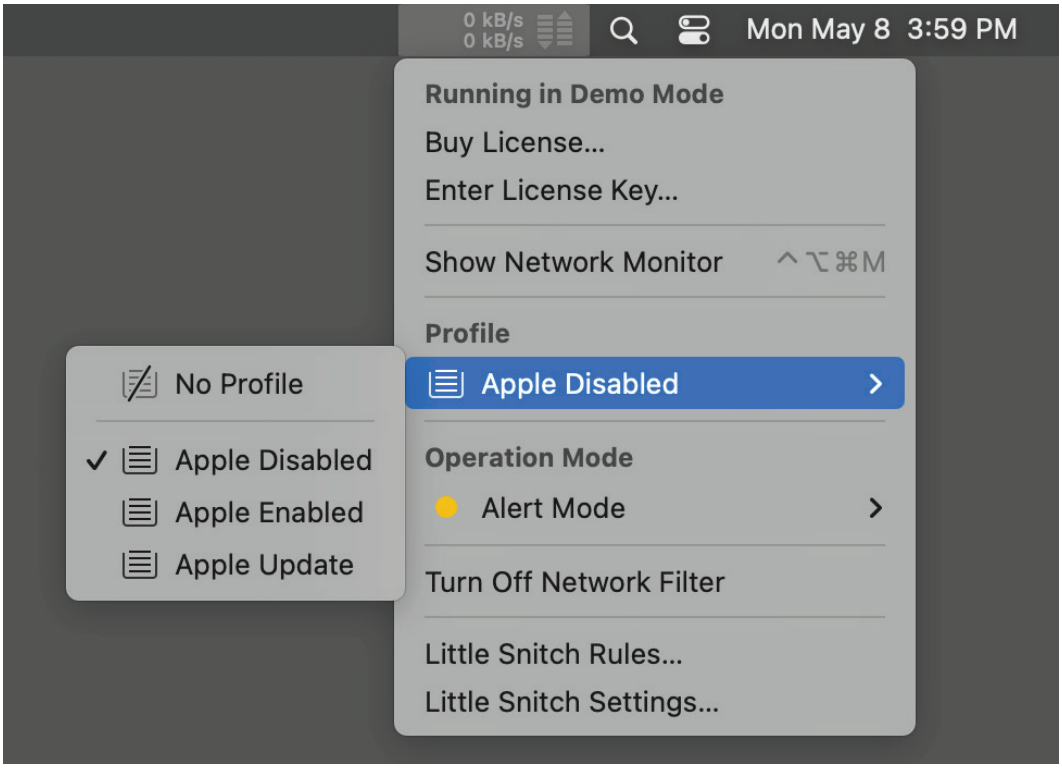
The DNS connections are required to translate domain names into IP addresses, which will be explained later. The time option allows our new time server to synchronize our clock. The Safari option allows the native web browser to connect to any website, which will be discussed later in this guide. The final option allows Little Snitch to periodically check for updates.

As you can see, initial setup of the firewall is time consuming and tedious. Making a mistake could prevent your computer from accessing the internet. You could create multiple profiles which would allow you to easily strengthen or weaken the settings, but that would be even more time consuming. This is why I recommend using my custom import file which will configure everything for you. The following should configure your copy of Little Snitch to perform just like mine.

- Open Safari and navigate to https://inteltechniques.com/data/LS.xpl.
- If prompted, "Allow" Safari to download the file.
- Click the Little Snitch menu icon and select "Little Snitch Rules".
- In the menu bar, select "File" and "Restore from backup".
- Click "Browse" and choose the "LS.xpl" file in "Downloads".
- Click "Open", "Next", and "Import".
- Enter your password if prompted and close the window.

Your instance of Little Snitch should now reload with my custom settings. Click on the Little Snitch menu icon and notice the changes. The following figure displays how it should appear in version 5. Version 6 will require you to click the blue menu icon next to "Alert". Notice that you now have three new profiles. Let's understand each.

- **Apple Disabled:** All core Apple connections are disabled.
- **Apple Enabled:** All core Apple connections are enabled.
- **Apple Update:** All core Apple connections are disabled except for those required by the operating system update process.



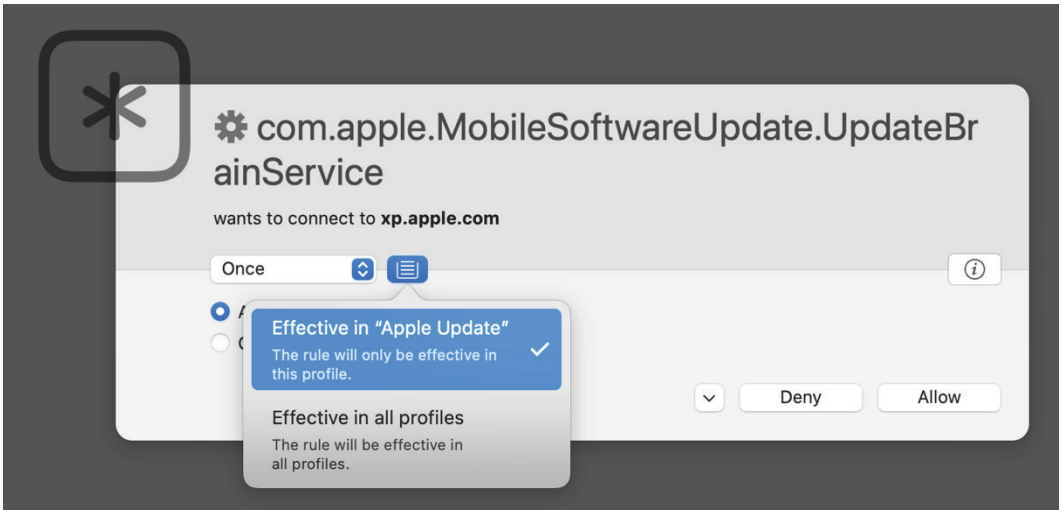
Now, let's test these profiles. Make sure you have selected the "Apple Disabled" profile and that Little Snitch is set to "Alert Mode", as seen in the previous image. Now, attempt to update your operating system with the following steps.

- Open "System Settings" and click "General" then "Software Update".

You should receive an error similar to "Unable to check for updates". Now, close the System Settings application. Switch to the "Apple Update" profile within Little Snitch and repeat the process, which should be as follows.

- Open "System Settings" and click "General" then "Software Update".
- Click "Update Now" and "Agree".
- Enter your password if prompted.

Your system should begin to install any pending updates, or may tell you that all updates have been applied. If Little Snitch prompts you to allow or block a connection similar to "com.apple.MobileSoftwareUpdate...", choose "Forever", Any Connection", and "Allow" into the "Apple Update" profile. The following image shows my settings when I allowed it. This service seems to be unique to each version of the operating system, and my setting may not carry over to yours.



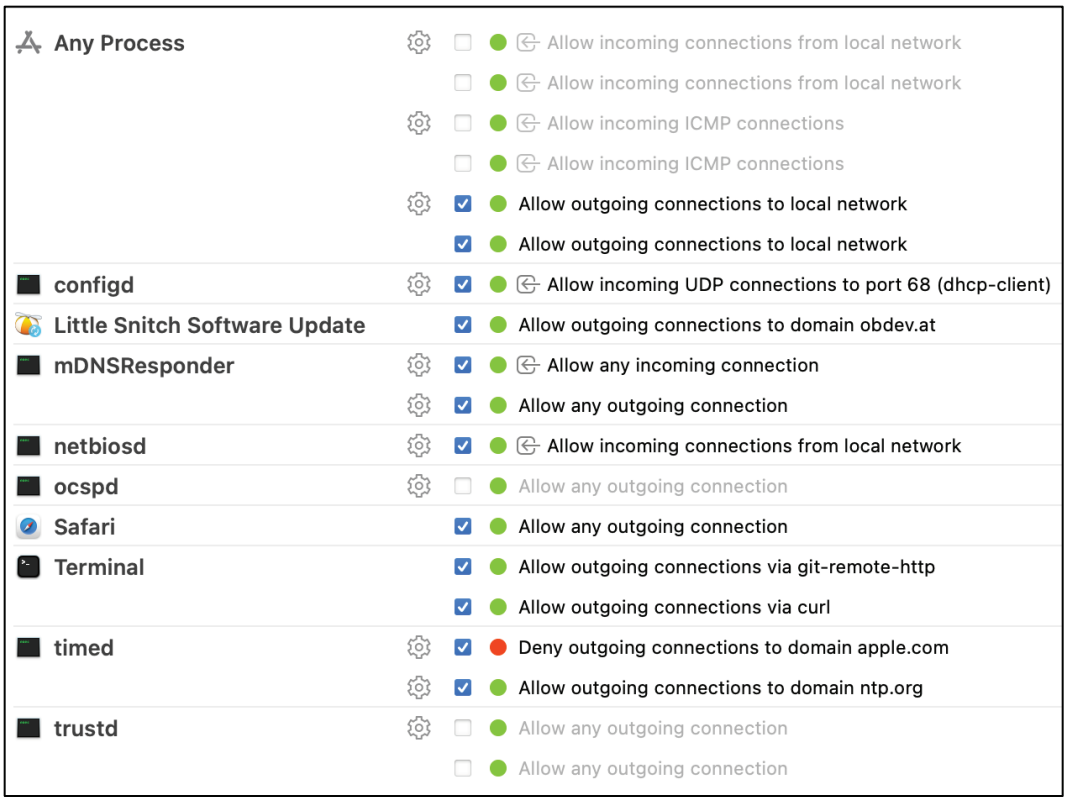
After your machine has fully updated and rebooted, consider the most appropriate level of firewall protection for your needs. Mine is always set to "Apple Disabled" and blocks all of the invasive actions by Apple. Once weekly, I switch to "Apple Update" and check for any pending updates using the previous tutorial. After updating, I switch back to "Apple Disabled". If I ever need to troubleshoot a connection or application issue, I could switch to "Apple Enabled". This allows every connection from Apple, so be careful. I never use this setting.

Let's dive deeper into the "Rules" behind each profile. Click the Little Snitch icon in the menu bar and select "Little Snitch Rules". In the left menu is a section labeled "Profiles". You should see the following four options.

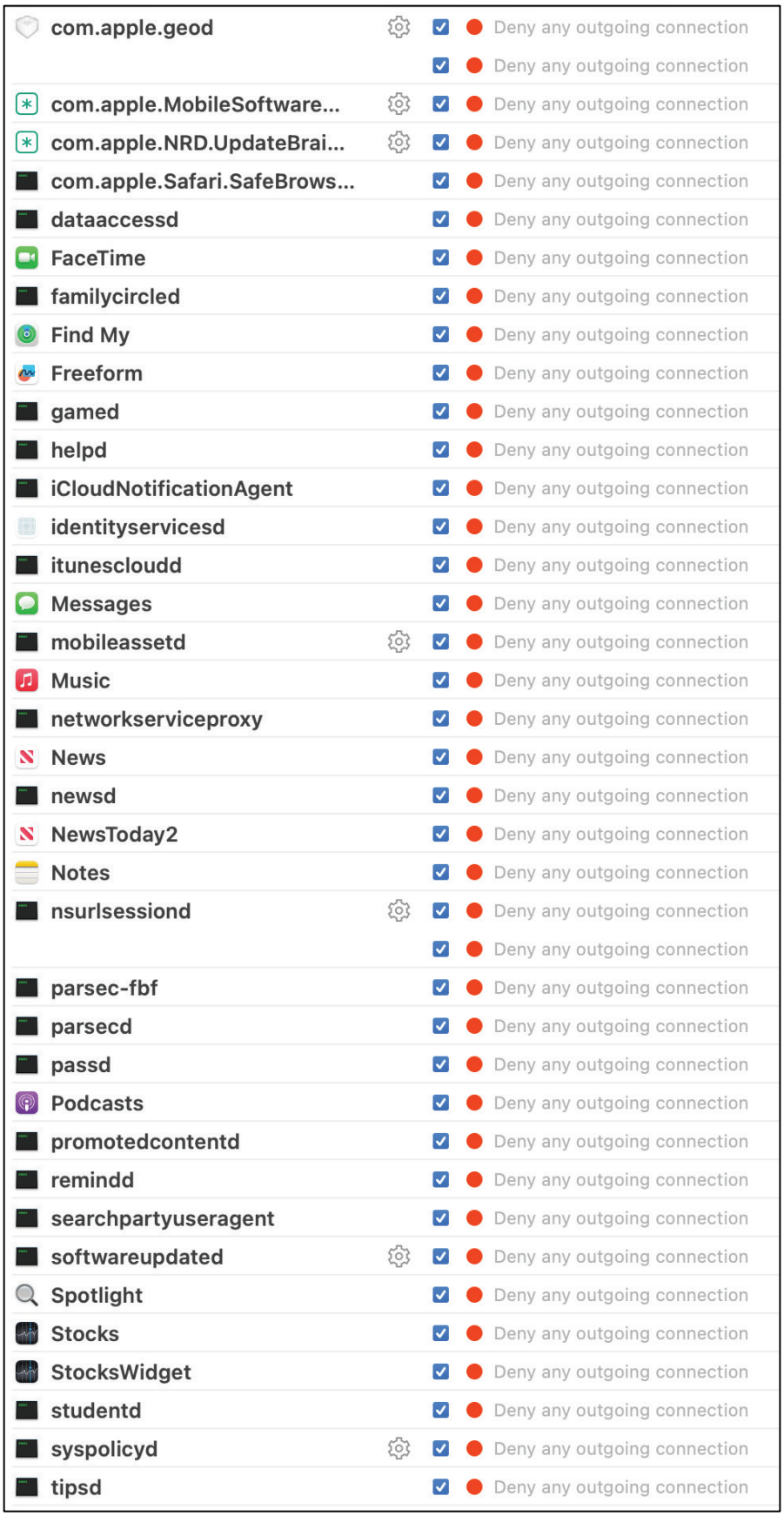
- **Effective in all profiles:** These rules will always be applied, regardless of which profile is active.
- **Apple Disabled:** These rules only apply when this profile is active, and all Apple services are blocked.
- **Apple Enabled:** These rules only apply when this profile is active, and all Apple services are allowed.
- **Apple Update:** These rules only apply when this profile is active, and only the services needed to update your machine are allowed.

Please note that these profiles will only be imported into Little Snitch for the current user. If you have multiple accounts on your macOS, you would need to replicate the import for any user desired. Little Snitch does offer a "Global Rules" setting which applies to every user, but I believe that is unnecessary for us.

The following image displays the current settings for the "Effective in all profiles" option. These allow connections within the local network, Little Snitch updates, your network's default DNS server, websites through the Safari browser, our new time server, and some basic Terminal connections. The unchecked boxes are required system settings which must be present within Little Snitch, but they are disabled.



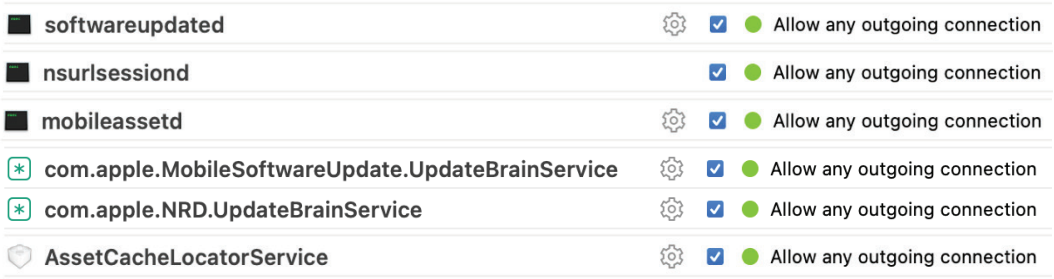
The next three images display the settings for the "Apple Disabled" profile. Notice that everything is blocked. These invasive Apple data collection endpoints are prevented at all times when this profile is selected.


com.apple.geod
Deny any outgoing connection
Deny any outgoing connection
com.apple.MobileSoftware...
Deny any outgoing connection
com.apple.NRD.UpdateBrai...
Deny any outgoing connection
com.apple.Safari.SafeBrows...
Deny any outgoing connection
dataaccessd
Deny any outgoing connection
FaceTime
Deny any outgoing connection
familycircled
Deny any outgoing connection
Find My
Deny any outgoing connection
Freeform
Deny any outgoing connection
gamed
Deny any outgoing connection
helpd
Deny any outgoing connection
iCloudNotificationAgent
Deny any outgoing connection
identityservicesd
Deny any outgoing connection
itunescloudd
Deny any outgoing connection
Messages
Deny any outgoing connection
mobileassetd
Deny any outgoing connection
Music
Deny any outgoing connection
networkserviceproxy
Deny any outgoing connection
News
Deny any outgoing connection
newsd
Deny any outgoing connection
NewsToday2
Deny any outgoing connection
Notes
Deny any outgoing connection
nsurlsessiond
Deny any outgoing connection
Deny any outgoing connection
parsec-fbf
Deny any outgoing connection
parsecd
Deny any outgoing connection
passd
Deny any outgoing connection
Podcasts
Deny any outgoing connection
promotedcontentd
Deny any outgoing connection
remindd
Deny any outgoing connection
searchpartyuseragent
Deny any outgoing connection
softwareupdated
Deny any outgoing connection
Spotlight
Deny any outgoing connection
Stocks
Deny any outgoing connection
StocksWidget
Deny any outgoing connection
studentd
Deny any outgoing connection
syspolicyd
Deny any outgoing connection
tipsd
Deny any outgoing connection

If you imported this configuration and look at the "Apple Enabled" profile, you would see that all of these same settings are green. This profile is the equivalent of disabling Little Snitch. However, it would allow you to keep blocking third-party programs while allowing all Apple services. I never use this profile, but I could see two situations where it might be valuable.

- **Temporary Apple Applications**: If you only occasionally need to use Apple services such as FaceTime and iCloud, the "Apple Enabled" profile would allow all features to function. However, I discourage this and hope you will embrace the alternative options within the next chapter.
- **Troubleshooting**: If you have an application, service, or entire operating system which seems to be non-functioning, you could temporarily allow all Apple connections to see if this corrects the behavior. You could also disable the firewall completely for even more thorough allowances. However, this immediately exposes your device's data to Apple. Again, I never select the "Apple Enabled" profile.

Finally, there is the "Apple Update" profile previously mentioned. It is a replica of the "Apple Disabled" profile, but includes the following allowed exceptions for the services required by Apple to update the operating system.



It is important to note that this custom configuration file for Little Snitch could break your desired apps. If you ever decide to start using iCloud, FaceTime, or any other Apple products, the "Apple Disabled" profile will prevent you from accessing these services. Therefore, you should understand how to modify these setting when needed. Let's assume that you want to block most of Apple services, but still want to use the stock Apple Podcast application. Within the Little Snitch Rules window, locate

"Podcasts" within the "Apple Disabled" profile. You could either delete it by right-clicking on the option, or double-click it to modify the setting. You could change "Deny Connections" to "Allow Connections", and would not need to reconfigure the setting the next time you open the Podcast application.

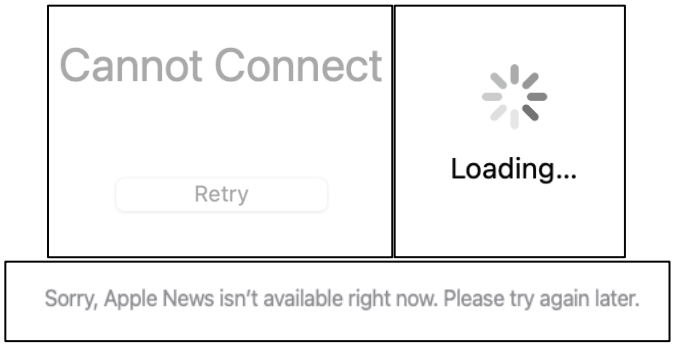
Important: If you disabled Gatekeeper within the previous chapter, you do not need to take any action within Little Snitch. The "Apple Disabled" profile blocks a process called "syspolicyd", which is the way Apple confirms allowed programs as part of the Gatekeeper service. If you did NOT disable Gatekeeper, and you want Apple to always confirm applications are approved when you launch them, you will need to allow "syspolicyd" within the rules.

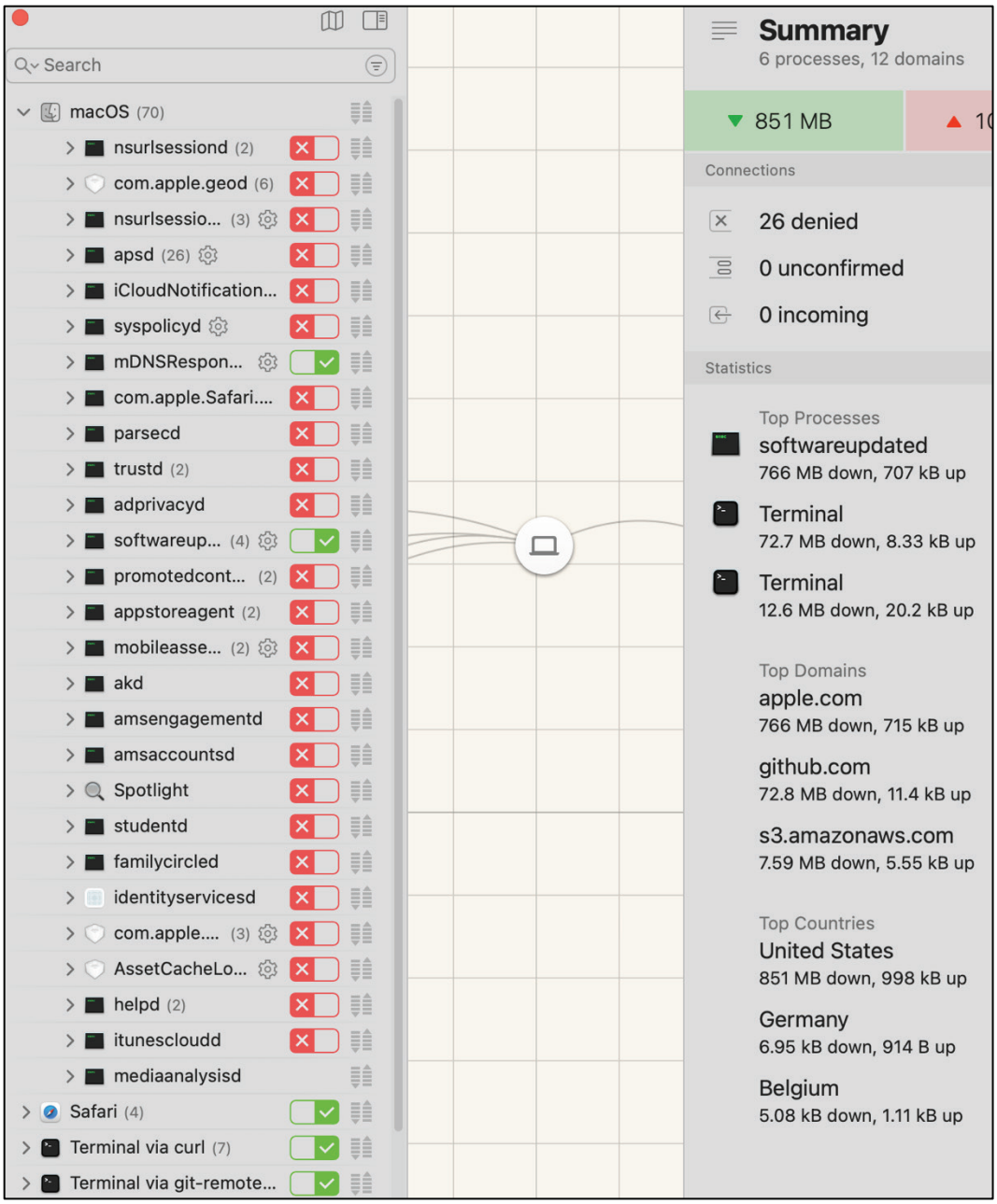
When we get into installing software within the next chapter, Little Snitch will prompt you any time an application requests to send data from your machine. Please note that this custom script only includes Apple stock applications, Homebrew, and FreeFileSync, but none of the remaining optional third-party software motioned throughout the rest of the book.

This application requires much time for proper configuration, but hopefully the pre-built import file will get you through the worst parts. Once configured for your needs, you will possess a more private operating system which shares much less data with Apple and other applications. As another example, I have my Mac set to block all outgoing connections to Microsoft when I open Word, Excel, or any other Office application. Microsoft does not need to be notified about my usage.

Let's take a look at our work in action. Click the Little Snitch logo in the upper menu bar of your system and select "Show Network Monitor". The image on the following page is a live example of my current configuration. I have expanded the "macOS" category, which displays all of the Apple connection attempts which were blocked. The only two which were allowed were my DNS service and the software update process. This is how you can monitor the ways in which this application protects your privacy. You may also notice that the map did not load within this window. This is because we are blocking system access to geolocation and Apple Maps.

Let's conduct another test. Execute the Books, Music, and News applications while the "Apple Disabled" profile is in use. You should see failures for each application. The following represents the result for those apps on my machine.





Even if you import my custom profiles, your work is not done. Apple occasionally introduces new telemetry or changes the details of their processes. Either of these will prompt you for approval of a new connection. If it is an Apple process which I do not want, I choose "Forever"; confirm in the dropdown that the rule will be applied to the "Apple Disabled" profile; click "Any Connection"; then "Deny". If I am in any other profile, I make sure the rule will be applied to that profile. I then enter that profile from the rules screen and copy and paste it to each of the other "Apple" profiles. I change "Deny" to "Allow" in the "Apple Enabled" profile in case it is ever needed. If the process is not from Apple, I choose my desired settings and make sure it is saved to the "Effective in all profiles" setting. In other words, I keep all settings for third-party software out of the Apple profiles. If I trust the app, I allow the connection.

As a reminder, a free trial of Little Snitch is limited to three hours during each boot. After that time, the software shuts off and you are exposed. If you reboot your computer every few hours, this may work for you, but it is not feasible for most users. I highly recommend purchasing this application, as it is affordable and provides a permanent license. I purchased my own copy under an alias name, and received nothing to promote this product. If you prefer a completely free and open-source software firewall, then Lulu is your best option, as explained next.

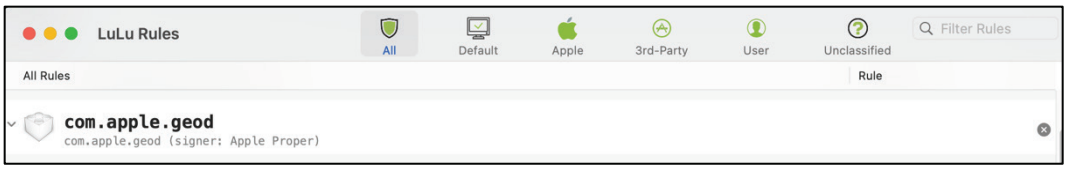
Lulu Configuration

A free alternative to Little Snitch is LuLu. Previously, I did not encourage readers to use this software as I believed Little Snitch was a much better product. I still prefer Little Snitch over LuLu, but the software has become quite a competitor over the past couple of years. You only need LuLu if you do not use Little Snitch, and it may be more desirable if you are on a limited budget. **Do not install both!** Lulu can be installed with the following steps.

- Download Lulu from https://objective-see.org/products/lulu.html.
- Double-click the file and drag the program into the Applications folder.
- Close the installation window.
- Launch the application.
- Click "Next" then "Open System Settings" when prompted.
- Click "Allow" in the "Privacy & Security" window then enter your password.
- Click "Allow" to permit Lulu to filter network content.
- Deselect all options within the Lulu configuration menu and click "Next".
- Choose whether you want to donate to the developer.

Lulu immediately prompted me to allow or deny a connection to Apple's location service, as seen below.

I chose the "Block" option, which was then added to the "Rules" window, as seen below.



Once LuLu is completely configured, it will be running and set to automatically start each time you log in. It will appear in the status bar in the upper-right of your desktop. LuLu aims to alert you anytime a new or unauthorized outgoing network connection is created with an alert containing information about the process attempting the connection. To approve an outgoing connection, such as from your web browser, simply click "Allow". To deny a connection, click "Block". Unless you click the "temporarily" button, a persistent rule will be created to remember your decision. By default, your decision to block or allow applies to the entire process.

Unfortunately, Lulu no longer offers an import and export feature, so I could not provide a pre-built settings file. While one could technically replace the "plist" file within Lulu's file structure to replicate this feature, I worry that doing so could cause other issues. If Lulu should ever re-enable import and export options, I will revisit the possibility with an update to this guide.

Overall, I hope you can see that Little Snitch is much more advanced and robust than Lulu. While both can effectively block undesired connections, Little Snitch allows more configuration, multiple profiles, and the ability to import settings. Both LuLu and Little Snitch have steep learning curves. However, once properly configured, they will silently protect you from eavesdropping apps.

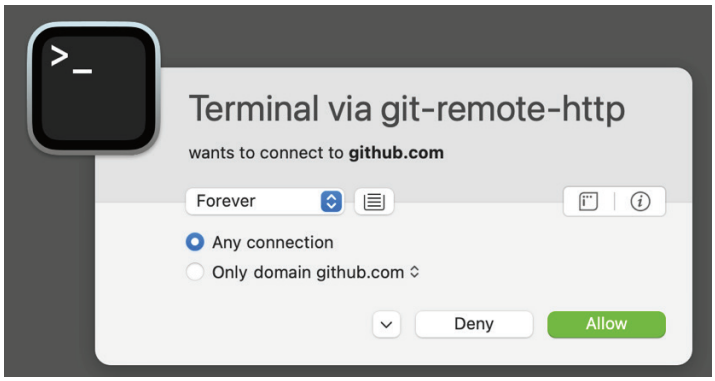
Now that you have your operating system properly configured and a firewall in place to protect you from data intrusions, we can finally start installing the applications which we will use on our new machine.

CHAPTER FIVE
APPLICATIONS

The first application I install on any new macOS operating system is a package manager called Homebrew, often shortened to Brew. This application is very beneficial when there is a need to install utilities which would usually already be present on a Linux computer. It also simplifies installation of applications which would otherwise require manual download or access to Apple's App Store. Brew is my favorite software for macOS computers. The easiest way to install Brew is to visit the website brew.sh then copy and paste the following command into the Terminal application (Applications > Utilities > Terminal). After completion, you are ready to use Brew to install and update applications.

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.
com/Homebrew/install/HEAD/install.sh)"
```

You will likely receive a notice from macOS that you need to install "Developer Tools". Click "Install" and "Agree", then allow the process to complete. If you adopted Little Snitch, as previously explained, make sure you are in the "Apple Update" profile, as the process is similar to any other macOS update. If manually following along, you will also likely be prompted by your chosen software firewall to allow or deny several Terminal connections to Github or other locations. These are the first non-macOS notifications we have received, but more are coming. The following displays my choice to "Allow" "Any Connection" "Forever" to Terminal. I had to do this a few times, but should not be asked again for this process. If you imported my custom Little Snitch rules, I have already authorized this connection on your behalf.



After Brew installation is complete, you will likely be presented with one or two commands which need to be manually executed within Terminal. My installation presented the following notice.

```
==> Next steps:
- Run these two commands in your terminal to add Homebrew to your PATH:
    (echo; echo 'eval "$(/opt/homebrew/bin/brew shellenv)"') >> /Users/ventura/.zprofile
    eval "$(/opt/homebrew/bin/brew shellenv)"
```

This is unique to my installation, as my chosen username was "ventura" at the time. Copy any commands presented here and paste them within the same Terminal window, executing each by striking return. Let's test everything with a few commands.

- `brew doctor` - This command confirms that Brew is configured properly and that all paths are set. You should receive a notice that "Your system is ready to brew".
- `brew update` - This command checks for any pending updates to Brew itself. You should receive a response of "Already up to date".
- `brew upgrade` - This command updates any installed programs. You should receive no response since we have not installed anything.
- `brew analytics off` - This command disables Brew's embedded analytics which monitor the number of times an application is installed using Brew. These metrics are only used to understand how users interact with the product, but I prefer to limit my exposure.

If everything is working, you are now ready to use Brew as a software installation repository. Treat this as a replacement for the App Store, but it does not require an Apple ID. Let's use it to install our first three applications.

TaskExplorer: This free macOS application is simple yet effective. It identifies all running processes and queries them through a service called Virus Total. If it finds a suspicious file, it alerts you with a red flag in the lower-right corner. Clicking the flag allows you to see more details about the potential threat. I execute this program weekly from any Mac machine I am using. If you have picked up a virus on your host, this program should identify it quickly. However, it does not remove any infections. For that, you will need to research any suspicious files. The following terminal command installs TaskExplorer to your Applications folder.

```
brew install taskexplorer
```

KnockKnock: Similar to the previous option, which is maintained by the same company, this program also conducts a scan of your Mac device. However, it is looking for persistent programs which are set to launch upon boot. Since most viruses inject themselves to launch the moment your computer starts, this program may identify threats which were missed by the previous program if they were not running at the time. After opening this application, click the scan button and allow the process to complete. You will receive a notification about any suspicious files. I execute this weekly along with TaskExplorer. Please note that it only notifies you of issues, and does not remove them. The following terminal command installs KnockKnock to your Applications folder.

```
brew install knockknock
```

Both TaskExplorer and KnockKnock never upload your files to Virus Total. They simply generate a unique hash of any system files and queries its website for any presence of that hash value. I trust this process. I no longer recommend macOS anti-

virus programs, such as the previously recommended ClamAV, since these two programs are more likely to identify anything malicious.

Onyx: If your Apple operating system is behaving strangely, Onyx may be able to correct the issue. This maintenance program should not be executed on a schedule, and should be reserved for situations of undesired behavior. On occasion, my fonts become corrupted and my menus become unreadable. Onyx fixes this. The following within Terminal installs Onyx.

```
brew install onyx
```

Next, I like to install various Terminal utilities which will be required for future tutorials. These are not programs which can be opened from the Applications folder. These are utilities which quietly wait until they are needed. I executed the following command within Terminal.

```
brew install bash coreutils curl ffmpeg grep ripgrep wget
youtube-dl yt-dlp
```

Replacement Applications

Computers running macOS have several default applications which function well and are beautiful. They are also full of privacy concerns. In the introduction of this book, I explained how Apple collected and stored sensitive information about me due to my usage of their Mail, Podcast, and Calendar applications. Now that we have blocked all of this invasive eavesdropping with our firewall, we need alternative options. Software selection is a very personal choice, and you may not agree with my recommendations. Therefore, I only present the following as examples of what has worked well for me and my clients as replacement for the stock Apple programs.

Email, Calendar, & Contacts

Apple's default Mail, Contacts, and Calendar applications are simple, effective, and well-designed. They are also collecting user information, including your contacts and appointments, and sending that data to Apple's servers. Some readers might have no need for a software client if they access their messages, events, and contacts through a web browser, but I believe everyone should have an offline backup of all this content on their computer. This is the only reason I launch my email client weekly. It downloads all new messages and other data onto my machine. I can then search them offline if needed, and I have a full back up in the event my provider has issues.

While I rely on **MailMate** (freron.com) for email due to its simplicity and bare-bones approach, many clients did not like the overall experience. Today, most of my clients possess **Thunderbird** (thunderbird.net) as their email, calendar, and contacts software. Thunderbird is free and open-source, but carries a stigma of being outdated and unusable. It has received several updates over the past year, and a huge overhaul is pending. As I write this, I am using version 102.11.0. In Summer of 2023, we expect to see version 115 "Supernova", which is a complete redesign for which I am eagerly

waiting. I believe it should be a true competitor to Apple's stock apps. Until then, we can still enter its ecosystem as a replacement for Mail, Calendar, and Contacts. Thunderbird can be installed with the following Terminal command.

```
brew install --cask thunderbird
```

After installation and execution of the software, your firewall should alert you of attempted connections. This will be common for any software installation, but I want to present one more example of how I use Little Snitch to protect me from unnecessary data collection. When I launched Thunderbird, Little Snitch immediately notified me that the new program was attempting to connect and send data to "location.services.mozilla.com". I did not want this to occur, so I chose "Forever", "Only domain mozilla.com", and "Deny". Notice I did not choose "Any connection" this time. This is because I will need Thunderbird to connect to my email provider soon, and I want to be prompted when this happens.

Next, I was prompted to connect to "www.mozilla.org", which I also chose "Forever", "Only domain mozilla.org", and "Deny". This is likely to send telemetry about a successful installation, which has no benefit to me. The pop-ups stopped, and Thunderbird seemed ready for use. I entered my email address and password into it, so that it could fetch my account information and was presented more notices from Little Snitch. The first was a request to use my DNS provider for DNS translation; the second was an attempt to query Thunderbird's servers for details about my email provider; and the third was to allow access to the email provider's domain. These were all acceptable, so I chose "Forever", "Only domain...", and "Allow". Thunderbird was now able to fetch my email without sending unnecessary data to Mozilla's servers.

Any time you install software which needs to make several connections to various servers, expect to see numerous notifications from Little Snitch. This is the way it is designed to function. It may seem overwhelming at first, but once everything is configured, these minor annoyances should stop. This level of granular control makes it possible to use macOS and any desired application in a way which grants you full control of your data. The time you spend now will pay off in the future. If you ever make a mistake, simply delete that entry within Little Snitch's rules and wait for the next notification when another attempt is made.

I do not want to present Thunderbird as an Apple Mail clone. It may appear outdated to you, but the function is all we care about. I never send emails from Thunderbird, or add contacts or events. I only use it as a way to retrieve and archive the data from my email provider. I rely on the web browser to connect directly to my provider for daily email, calendar, and contact access. The backup may never be needed at all.

My personal email accounts are all through Proton Mail, which includes my email, contacts, and calendars. I use the Proton Mail Bridge application to allow my email client to fetch and archive my messages via IMAP. On occasion, I manually export my calendar from Proton as an ICS file and manually import it into Thunderbird. I export my email contacts as a VCF file and manually import them into Thunderbird.

Stock Calendar and Contacts Applications

If you miss the stock Apple Calendar and Contacts applications, you do have a secure option for that type of sensitive data. EteSync (etesync.com) is a service which offers end-to-end encrypted calendar, contacts, and note data. For $2 monthly, they will store your data on their servers in a secure way in which they cannot access any of your data. You can then download the EteSync Bridge software (github.com/etesync/etesync-dav/releases) onto your macOS device. This software connects your encrypted EteSync data to the stock Calendar and Contacts applications. Any changes made within these apps are immediately reflected within the EteSync data. You could also synchronize this data with any other device, such as a smart phone. The one-time setup can be a hassle, but should never need repeated. If you insist on using Apple's applications for this type of data, EteSync is your best option. Unfortunately, there is no Homebrew installation option, and the current version only works on Intel-based machines. You will need to refer to EteSync's online documentation.

Notes

I am a huge fan of **Standard Notes** (standardnotes.com). The free version provides fully end-to-end encrypted (E2EE) data. Only you can see your content, and you can synchronize all data to any other desktop or mobile device. I rely on my notes throughout every day. The paid version introduces more text formatting options and spreadsheet entries, but I prefer the plain-text feel of the free edition. However, the paid edition includes the ability to store two-factor authentication (2FA) codes, which is a huge benefit. This allows me to possess a truly cross-platform, open-source, encrypted application for my 2FA. Standard Notes can be installed with the following.

```
brew install --cask standard-notes
```

Books

I prefer Calibre (calibre-ebook.com) as my eBook library software. It allows me to collect the books I have purchased or downloaded and synchronize them to practically any eBook reader. It can be installed with the following Terminal command.

```
brew install --cask calibre
```

TextEdit

The default TextEdit application within macOS is riddled with problems. By default, it opens files in rich-text format; automatically corrects any spelling it deems appropriate; and appends file extensions when unnecessary. All of these annoyances can be corrected within the settings, but I prefer to ditch it altogether and use something better. I currently use and recommend the free version of **BBEdit** (barebones.com/products/bbedit). This is a true text editor with many advanced features, but it leaves them out of your way unless you need them. We will use it many times later when we start making our own scripts. It can be installed with the following Terminal command.

```
brew install --cask bbedit
```

Messages/FaceTime

Apple wants you to use their own proprietary software and services for all voice, video, and text communications. I believe this should be avoided. At the risk of stepping outside the scope of this book again, I strongly encourage you and your contacts to adopt encrypted (E2EE) messaging for these tasks. Both **Signal** (signal.org) and **Wire** (wire.com) provide open-source E2EE platforms for voice, video, and text. There are many other great options, but these are the best for those who are new to the idea. Each can be installed with the following Terminal commands.

```
brew install --cask signal
brew install --cask wire
```

Password Managers

Apple includes its own password manager within Keychain, but I do not recommend it. It is not cross-platform and might be inaccessible if you lose your machine or experience hardware failure. I only recommend open-source encrypted password managers which can easily export a backup for use on another machine. I believe every reader of this book should possess a password manager, and I recommend two options for two unique experiences.

Offline - KeePassXC (keepassxc.org): If you want extreme privacy and cannot tolerate the idea of storing your passwords in the cloud, regardless of the encryption and security, then an offline password manager is appropriate for you. I use KeePassXC and make sure I have a good backup. It can be installed with the following Terminal command.

```
brew install --cask keepassxc
```

Online - Bitwarden (bitwarden.com): If the idea of manually synchronizing your password database to any other device or backup drive makes you want to give up on the security benefits of a password manager, then Bitwarden is the best option. It is much easier than KeePassXC and keeps all of your passwords updated across devices, but it also technically stores an encrypted version of your passwords on the internet. This data is protected with strong encryption and not even Bitwarden employees can access your data, but you are presenting a slight risk. It can also handle all software token 2FA while itself being protected with a hardware Yubikey. Most of my clients choose Bitwarden while a few extremists go with KeePassXC. Bitwarden can be installed with the following Terminal command.

```
brew install --cask bitwarden
```

Music

I refuse to use the default macOS Music application, which attempts to collect information about your interests and send it to Apple's servers. Many people are satisfied with a standalone media player such as VLC, but I prefer more features. I currently rely on **Swinsian** (swinsian.com) as my complete music library solution and **Mp3tag** (mp3tag.app) as an advanced file tagging application. These allow me to possess my own copy of my music without relying on streaming services. Keeping all audio files properly tagged makes my collection organized and efficient. Neither of these programs are free, but both offer a free trial to see if they are right for you. I believe I have tested all music library applications for macOS, and Swinsian was the only one which performed similar to Kodi, which I use as my home media server. Kodi offers a free macOS version, but I find it to have appearance issues when used in windowed mode (Kodi is designed to be used as a full-screen media player). Mp3tag was the only macOS tagging application I found which allows access to legacy metadata which was preventing my collection from being properly organized.

Application Updates

You should keep your newly-installed software updated. The "Software Update" options within "System Preferences" will patch your operating system, but it does not update individual applications. Since we used Brew to install our optional software, the following commands will update Brew itself; update each application; force an update of any older versions; cleanup any unnecessary cached files; remove the Homebrew cache itself; replenish any missing dependencies; remove software no longer needed by your computer; and check that everything is configured properly. I keep these commands digitally ready within my local notes application for easy copying and pasting, but a future chapter will present a custom script which will automatically execute all of these commands.

```
brew update
brew upgrade
brew upgrade --greedy
brew cleanup -s
rm -rf "$(brew --cache)"
brew missing
brew autoremove
brew doctor
```

You should now possess a macOS computer which is stable and secure, and includes the basic applications for daily use. There is still much more to be done, but you have the staples completed. If you used Brew to install all of your applications, you do not need any of the popular "App Cleaner" style of programs. Once you remove a program with Brew, the previous steps also clean up the remains. When we create our custom maintenance script later in the book, we will add even more commands to make sure we are keeping our system tidy. Next, we must consider proper web browser configuration and usage.

CHAPTER SIX
WEB BROWSERS

Before we consider connecting to various websites in order to take full advantage of our new macOS build, we should configure our web browsers. I recommend the Firefox web browser for most daily browsing, with limited use of Apple's native Safari browser for specific tasks. Safari is quite secure and private by default, but it does not allow the level of configuration which Firefox presents us. We will use both, but let's begin with Firefox. You can install Firefox with the following Terminal command.

```
brew install --cask firefox
```

Once installed, execute the application and consider the following modifications.

- Click on the Firefox menu in the upper right and select "Settings".
- In the "General" options, uncheck "Recommend extensions as you browse" and "Recommend features as you browse". This prevents some internet usage information from being sent to Firefox.
- In the "Home" options, change "Homepage and new windows" and "New tabs" to "Blank page". This prevents Firefox from loading their default page.
- Disable all "Firefox Home Content" options.
- In the Search options, change the default search engine to DuckDuckGo and uncheck all options under "Provide search suggestions". This prevents queries from going directly to Google, and blocks the Google API from offering search suggestions.
- Uncheck "Browsing history" from the "Address Bar" menu.
- Click the "Privacy & Security" menu option and select "Strict" protection.
- Added settings for Firefox's "Global Privacy Control" and "Do Not Track".
- Check the box titled "Delete cookies and site data when Firefox is closed".
- Uncheck the box titled "Show alerts about passwords for breached websites".
- Uncheck the box titled "Suggest and generate strong passwords".
- Uncheck the box titled "Autofill logins and passwords".
- Uncheck the box titled "Ask to save logins and passwords for websites".
- Uncheck the boxes "Autofill addresses" and "Autofill credit cards".
- Change the History setting to "Firefox will use custom settings for history".
- Uncheck "Remember browsing and download history" and "Remember search and form history".
- Check the box titled "Clear history when Firefox closes". Do not check the box titled "Always use private browsing mode", as this will break Containers.
- In the Permissions menu, click "Settings" next to Location, Camera, Notifications, and Virtual Reality. Check the box titled "Block new requests…" on each of these options. If you will never need audio communications within this browser, you could do the same for Microphone.

- Uncheck all options under "Firefox Data Collection and Use".
- Uncheck all options under "Deceptive Content and Dangerous Software Protection". This will prevent Firefox from sharing potential malicious site visits with third-party services.
- Select "Enable HTTPS-Only Mode in all windows".

These settings are what I refer to as the basics, and may be enough for most readers. This is where I want to deviate from previous writings. Prior to this publication, I always presented several settings which could be modified within the Firefox "about:config" menu. I no longer recommend this, which may upset some privacy fanatics. Please consider my reasons before abandoning this chapter. The purpose of my previous recommendations was mostly to prevent browser fingerprinting or canvasing. This activity is often abused by online sites which try to track you as you navigate the internet. If I own a clothing website and I can collect numerous identifiers from your browser, you are unique from everyone else who has ever visited my site. When you come back a week later, I know you are the same user, and I can continue to track your activity within my site.

Previously, I had recommended readers change various settings, such as the ability for a website to know your current battery level, in order to make you appear less unique to the malicious systems snooping on your connection. Today, thanks to advancements in fingerprinting technology, I believe those settings could do more harm than good. If you are the only visitor on that site which disabled this setting, you now appear even more unique than if you had done nothing.

At the risk of offending some readers, I firmly believe the following statement. **We can no longer defeat modern browser fingerprinting by making modifications to our browsers.** Anything we change, or any extension we add, almost always makes us a unique visitor in the eyes of sophisticated fingerprinting systems. The more actions you take to blend into the crowd likely makes you stick out more. There are exceptions to this, but for general usage, sites will continue to track us. They will also collect our IP addresses, installed fonts, video characteristics, location metrics, and browser specifications at all times. That will never change.

Furthermore, many of the current recommended about:config Firefox privacy tweaks break other desired functions. If you try to block all possible IP address leakage via WebRTC, you will likely also break the ability to use voice and video conferencing within your browser. There must be a balance of protections versus functionality.

Using a VPN, as explained later in this chapter, and the previous Firefox settings stop most invasions. Since Firefox does not share cookies from one domain with another, we have strong privacy be default. If you want pure anonymity, stay off the internet. If you need the most protection possible, consider the Tor Browser, as explained later. Next, I will discuss the abundance of helpful browser extensions called add-ons.

The first vital add-on I install on every computer is **uBlock Origin**. It blocks many ads and tracking scripts by default, but it also can block any other type of script that is

attempting to run on a page. This helps prevent tracking, malicious code execution, location sharing, and a number of other processes that could undermine your privacy and security.

This add-on is completely free and open source. It is highly customizable, while remaining relatively easy to work with. uBlock Origin works from blacklists which block trackers specified in the list(s). The add-on comes with several lists enabled, but there are several more that can be added through simple checkboxes in the preferences. Keep in mind that the more blacklists you enable, it may be more difficult to work within the browser. This section may seem a bit overwhelming at first, but experimenting with the advanced settings should help you understand the functionality.

I have previously recommended NoScript, Adblock Plus, Privacy Badger, and Disconnect as privacy add-ons that would help stop unwanted ads, tracking, and analytics. These are no longer present on any of my systems. I now only use uBlock Origin, as it replaces all of these options. Let's start with the basics.

Install uBlock Origin from the Firefox Add-ons page or directly by navigating to the application's website at https://addons.mozilla.org/en-US/firefox/addon/ublock-origin/. You are now protected on a basic level. By default, most known invasive advertisements, tracking code, and malicious content is blocked. This step alone would provide much needed protection from the internet. However, we can take it a step further.

Click on the uBlock Origin icon in the menu and select the "Dashboard" icon to the right, which appears as a settings option. This will open a new tab with the program's configuration page. On the "Settings" tab, click the option of "I am an advanced user". This will present an expanded menu from the uBlock Origin icon from now forward. Click on the "Filter List" tab and consider enabling additional data sets that may protect your computer. I find the default lists sufficient, however I enable "Block Outsider Intrusion into LAN" under "Privacy" and the entire "EasyList" section under "Annoyances". Click "Update Now" after you have finished your selections. You now have extended protection which will be applied to all visited websites without any interaction from you. When you encounter a web page with a lot of advertisements, such as a news media website, it should load much faster. It will block many of the pop-ups and auto-play media that can be quite annoying when conducting research.

After you have enabled the Advanced settings as explained above, clicking on the uBlock Origin icon should now present an expanded menu which will change as you visit different sites. In order to explain the function of this menu, I will conduct a demonstration using the website cnn.com. Figure 6.01 displays the default view of uBlock Origin with the site loaded. Scrolling down this list of scripts that have either been loaded or blocked, you can see several questionable scripts such as Twitter, Amazon, and Turner. These scripts allow tracking across multiple websites and are the technology responsible for monitoring your interests, web history, and shopping habits.

This menu is split into three columns. The first simply identifies the type of code or domain name of the script. The second column is global settings. Anything changed here will apply to all website visits. The third column contains settings for the current website. A single plus sign (+) indicates that less than ten scripts were allowed from that specific option. Two plus signs indicate that between ten and one hundred scripts were allowed. The single minus sign (-) indicates that between one and nine scripts were blocked from that domain, while the dual minus signs tell us that ten to one hundred scripts were blocked.

In Figure 6.01, we know that over ten scripts were allowed to run from cnn.com, and at least one script was blocked from sending data to Twitter. This is all default behavior and provides a balance of functionality and security. uBlock Origin decides which content should be allowed and which should be blocked.

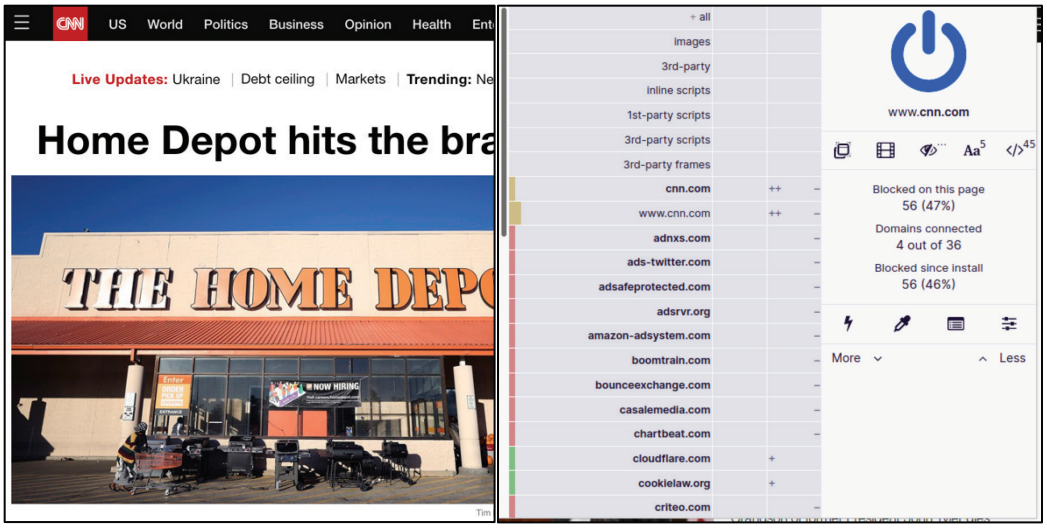


Figure 6.01: An advanced view of uBlock Origin.

Using this same page, let's modify the options. In Figure 6.02 (left), I have clicked on the far-right portion of the first cell in the third column. This turned the entire third column red in color. This action activated an option to refresh the page (arrows) and an option to save the change (padlock). Clicking the padlock and then refreshing the page presented me with the example in Figure 6.02 (right). Since I blocked every script, the page would not fully execute. It could not load images, design scripts, or any JavaScript. This is not useful at all, so I disabled my actions by clicking on the left (grey) section of the top cell in the third column, which turned the entire column back to grey in color. Saving these changes and refreshing the page brought me back to the example in Figure 6.01.

We can also take this to the opposite extreme. In Figure 6.03 (left), I clicked on the "power button" in the upper-right. This turned the entire left edge green in color, and allowed all scripts to load on cnn.com. This includes the dozens of intrusive scripts that could load advertisements on the page. You can also see that small plus signs

confirm that scripts were allowed to run while the minus signs in Figure 6.03 (right) state the opposite. For most users, this allowance would seem irresponsible.

Next, we will modify the second (middle) column, which will apply settings globally. By default, all options are grey in color, which is desired by most users. This indicates that the default block list is applicable, and only invasive scripts will be blocked everywhere. For demonstration, I clicked on the right (red) portion of the top cell in the second column. This turned the entire column red, and indicates that all scripts across all websites will be blocked. After I saved my changes, every website will only load the most basic text content. This will prohibit much of our usage.

Loading a page such as a Twitter profile resulted in no usable content. By clicking on the uBlock Origin icon and clicking the left (grey) sections of specific cells within the third column, I enabled those scripts without allowing everything on the page. In Figure 6.03 (right), you can see the difference in colors. In this example, the entire second column is red. This indicates that all scripts are blocked globally. The third column is mostly red, but the options for twitter.com and twimg.com are grey. Those scripts will be allowed, if approved by uBlock Origin's rules, only for that domain. If I load a blog that has scripts from Twitter, they would still be ignored.

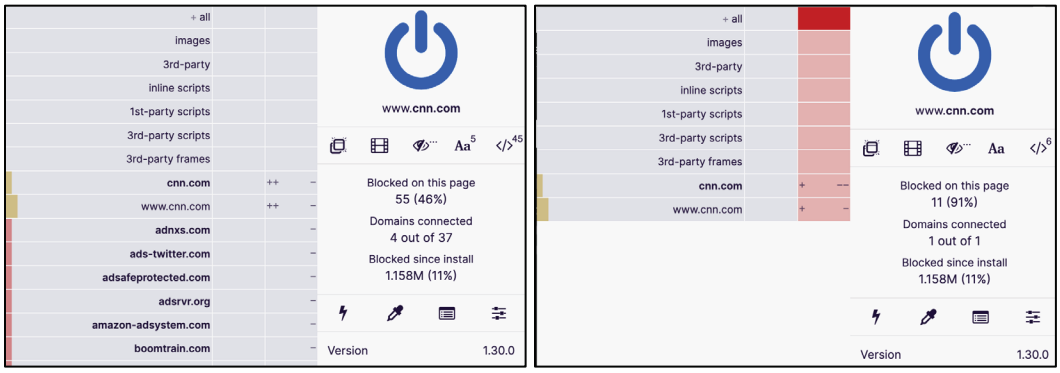


Figure 6.02: Disabled scripts within uBlock Origin.

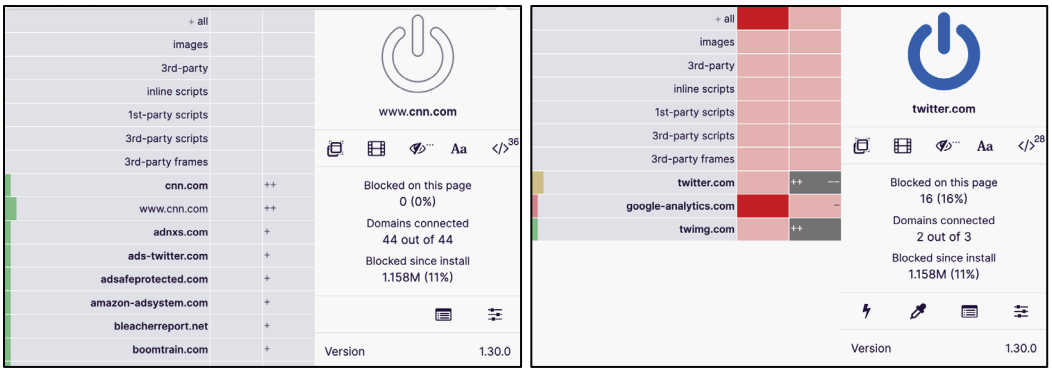


Figure 6.03: Fully and partially enabled scripts within uBlock Origin.

These are extreme examples. Let's bring this back to some sanity. The following is how I recommend using uBlock Origin. Install, enable advanced options, and proceed with your work. When you arrive at a website that is blocking something you want to see, open the menu and click on the left (grey) section of the top cell in the third column. That will allow everything to load on that page, and that page only. When you are about to navigate to a questionable site that may try to install malicious code on your machine, click on the right (red) section of the top cell in the second column. That will block all scripts on all pages. Conduct your usage and reverse the change when you are finished. Remember to click the save button (padlock) after each change and refresh the page.

Hopefully, you are practicing these settings and learning how this program functions. It is an amazing option that has protected me many times. If you are doing things right, you have likely completely messed-up your settings and are now blocking things you want while allowing things you do not. Don't worry, we can reverse all of our mistakes by first changing the global (second column) settings back to grey (left section of top cell). Next, return to the dashboard settings of the add-on, and click on the "My Rules" tab. In the second column (Temporary Rules), select all of the text and press the delete key on your keyboard. Click the "Save" button in this same column and then the "Commit" button to apply these settings everywhere. This resets our extension and brings us back to default usage regardless of your modifications. This is important in the event you go too far with settings in the future. Removing and reinstalling the extension does not always wipe this data out of your system.

The primary benefit of uBlock Origin over other options is the simple ability to block malicious scripts without customization, while having an option to allow or block any or all scripts at our disposal. This is a rarity in these types of add-ons. Another benefit is the ability to bypass website restrictions, such as a news site blocking articles unless the visitor has a subscription service. Consider the following example with the Los Angeles Times. Visiting the page allows you to view three articles for free, but you must have a paid subscription in order to continue using the site. If I click on the uBlock Origin menu while on this page, select the right (red) option on the right (third) column under the setting for "3rd party scripts", then the padlock icon, and reload the page, I see a different result. I am now allowed to see the article. This is because this website relies on a third-party script to identify whether a visitor is logged in to the service. This modification presents unlimited views of articles without registration on this and thousands of other websites.

The next Firefox add-on which I use daily is the **Multi-Account Containers** option from Mozilla. It can be found at addons.mozilla.org/firefox/addon/multi-account-containers. Prior to 2021, I used this service to create individual containers which isolated website cookies from each site. However, Firefox introduced "Total Cookie Protection" within version 86 released in February of 2021. Because of this, temporary internet files from each domain are confined to the websites where they originated (when "Strict" is selected under "Enhanced Tracking Protection"). Firefox creates a virtual container for each site loaded. Facebook cannot see the cookies downloaded from Amazon and vice-versa. Many believe this eliminates the need for Multi-Account Containers, but I disagree.

Multi-Account Containers allows you to separate your various types of browsing without needing to clear your history, log in and out, or use multiple browsers. These container tabs are like normal tabs except that the sites you visit will have access to a separate slice of the browser's storage. This means your site preferences, logged-in sessions, and advertising tracking data will not carry over to the new container. Likewise, any browsing you do within the new container will not affect your logged in sessions, or tracking data of your other containers.

Consider an example. I have a container tab open which I use to log in to a Twitter account. I want to log in to another Twitter account within the same browser. If I open a new tab and go to twitter.com, I am automatically logged in to the same account as the previous tab. However, if I open a new container tab, I am presented the option to log in to a new Twitter account. I simply open a unique container tab for each of these events. Each sees the session as unique, and no data is shared from one service to another. Once installed, you will see a new icon in the upper right which appears as three squares. Click on it and select the container you want to open. Default options include choices such as Personal and Shopping, but you can modify these any way you desire. I have ten containers titled Private01 through Private10. You can create, delete, and edit containers from the Containers menu. When you click the Edit Containers or the + buttons, you can change the color or icon associated with a container or change the container name.

I also use this extension in order to have quick access to all of my Google Voice numbers. I created a new container for each Google Voice number I own. I then logged in to the appropriate account for each container and disabled the option to clear my cookies upon exit. Today, I can launch Firefox, select the container titled with the number I want to use, and immediately place or accept a call via my desktop. I can close the browser completely when I am done. I also changed the icon and name to reflect this purpose. This has been most beneficial when I have been on a call with a financial institution and they want to call me back at a specific number which they have on file. Opening the browser and being immediately ready is better than connecting to Google Voice; opening my password manager; inserting my credentials; providing 2FA; accessing the account; allowing my microphone; and accepting the call. My device is encrypted and protected with a strong password in the event it is stolen.

Some readers may be frustrated with my setup for Firefox and may insist on using a Chromium-based browser. I completely respect this, and offer the option of Brave Browser. Brave is based on Chromium, which is the bones of the Google Chrome browser. Brave insists they have removed all calls to Google which Chromium makes by default, implementing the use of Quad9 as the DNS provider (instead of Google). However, Brave has faced strong criticism for injecting code to hijack affiliate web links and their overall push to use their embedded rewards program. If you NEED a Chrome-like browser, I recommend Brave over Chrome. If you can use Firefox, I find it to be much more privacy-focused. Personally, I would never use any Chromium-based desktop browser, including Brave.

KeePassXC-Browser

If you installed the KeePassXC password manager and want to access auto-fill functionality within Firefox, install the official extension from the Firefox repository on their site at https://addons.mozilla.org/en-US/firefox/addon/keepassxc-browser. After installation, conduct the following.

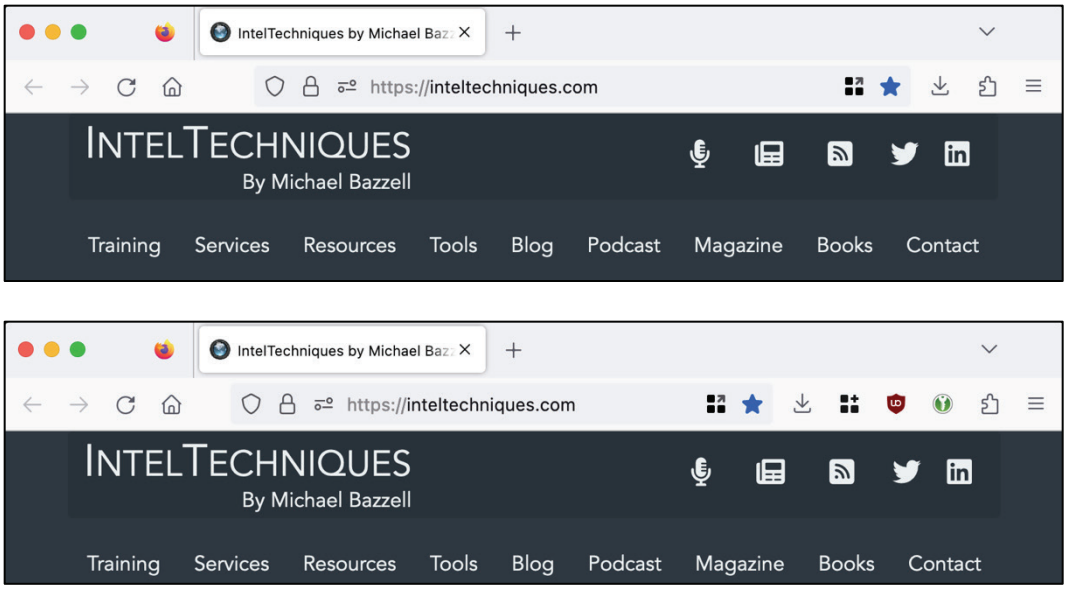
- Open your KeePassXC database and click the settings icon.
- Choose the "Browser Integration" menu option.
- Select "Enable browser integration and enable "Firefox".
- Click "OK" and return to Firefox.
- Click the KeePassXC icon and then click "Connect".
- Provide a unique name, such as "Firefox", and click "Save and allow access".

Your Firefox browser can now access passwords stored within the KeePassXC database without any data being sent over your internet connection.

Missing Icons

Firefox made a change recently which hides all extensions within the "Extensions" icon in the toolbar. I don't like this and prefer immediate access to my extensions. The following two images display the default configuration (top) and the appearance after modification (bottom). I conduct the following within Firefox.

- Click the puzzle piece in the upper-right.
- Right-click each entry and select "Pin to Toolbar".
- Right-click the toolbar and select "Customize Toolbar".
- Drag away any unwanted options and reorganize as desired.
- Click "Done" in the lower-right.



Safari

While I prefer a hardened Firefox browser for all common tasks, I recommend an untouched version of Safari for some specific purposes. Some websites, especially those related to the financial world, prohibit access from hardened browsers. If you try to log in to your bank website from a new browser with script blocking and no previously-stored cookies, you may be blocked. This is to prevent fraud and eliminate access by criminal groups. Unfortunately, we get caught up in these wide nets.

This is why I encourage clients to use Safari for some browsing. I believe readers should consider using Safari to access websites associated with their banks, investment firms, credit cards, and other scrutinized activity. Basically, consider Safari for any website which will be overly scrutinous of your login activity. This will prevent many blocks. Safari will keep a history of your access and store temporary internet files related to your login, but these can be beneficial when financial companies want to see that history of access.

In short, Safari can be great for tasks which must be overtly associated with your true identity, and a hardened version of Firefox is more appropriate for all other browsing.

Tor Browser

If you follow various privacy-related communities, you will hear many people promote the Tor Browser. This Firefox-based browser only connects to the internet via the Tor network, which hides your true IP address without a VPN. Since every Tor Browser user has the exact same browser configuration, we may not appear as a unique user when a website tries to fingerprint us. However, this is not perfect. If you are the only Tor Browser visitor to my fictitious clothing website, you stick out. Furthermore, many websites block connections from the Tor network altogether. If you try to log in to your bank from within Tor, expect to get blocked. Even worse, your account may be suspended.

I only use the Tor Browser for investigations which rely on it in order to access a specific website within the Tor network (which usually ends in ".onion"). For everything else, I rely on Safari or my hardened Firefox, as previously explained. You can install the Tor Browser with the following Terminal command.

```
brew install --cask tor-browser
```

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 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. Navigate to **inteltechniques.com/vpn.html** for further information and the best affiliate purchase links. Purchases include unlimited use, connection to multiple devices simultaneously, and fast speeds. I pay for my VPN with Bitcoin in an alias name, but that may be overkill for many readers.

For most readers, and almost every client I have consulted, I recommend sticking with the standard macOS application provided by the VPN company. These branded apps should suffice for most needs. Proton VPN can be downloaded with the following Terminal command.

```
brew install --cask protonvpn
```

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.

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 macOS device, I am connected through my VPN. I rely on Proton VPN through their app on my macOS device only while I am traveling. Home devices are protected through a firewall with Proton VPN, as explained in ***Extreme Privacy, 4th Edition***. At home, I never need to launch a VPN within my computer itself due to the VPN firewall.

I trust Proton VPN 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 and configurations will be posted on my website at **inteltechniques.com/vpn.html**.

Relying on a VPN company is difficult. We place a lot of trust into the provider(s) we choose, without knowing much about the parent companies. One could argue that the huge parent companies might have ill intentions for the data collected from millions of VPN users. One could also argue that being a small needle within their huge haystack might possess its own benefits. I believe all VPNs are flawed, but still a requirement for us. Almost every VPN provider relies on rented servers across the globe which are out of their control. Some providers unknowingly use the same servers as their competition.

When using a VPN, you are simply placing your internet history into someone else's hands. This sounds bad on the surface, but it is better than doing nothing at all. Without a VPN, we know our ISPs are monitoring, collecting, and sharing our internet activity. With a VPN, we are told that this information is not logged or shared. Are we bullet-proof? No. However, I would rather make the attempt to hide my traffic than do nothing at all. **My main purpose for a VPN is to prevent services such as my email provider from knowing my true home IP address.** Your needs may differ.

Some may question the amount of data shared about your online history when you send all of your traffic through a VPN versus your ISP. There are always vulnerabilities which could expose more data than intended, but we can discuss a few misconceptions about your internet traffic. First, we should tackle SSL/TLS. SSL (Secure Sockets Layer) and its successor, TLS (Transport Layer Security), are protocols for establishing authenticated and encrypted links between networked computers. This is related to the lock icon you see in your browser address bar when on any website which begins with "https". This indicates a secure connection, but what does that really mean? I will simplify this technology with a couple of examples.

Assume you are on your home computer connected directly to your internet service provider (ISP). You are not using a VPN. You connect to Google and conduct a search for "inteltechniques". The response URL presented to you, including the search results from the query, is https://www.google.com/search?q=inteltechniques. Does your Internet Service Provider (ISP) know you conducted a search on Google? Yes. Do they know you searched for "inteltechniques"? No. This is because Google encrypts the actual search URL. The provider of your internet connectivity can only see the domain name being accessed. It cannot see any details about specific pages or any credentials entered.

This is why https versions of websites are so important. Your browser can see this entire URL, but it does not directly share any details with your provider. Now, let's introduce a VPN. After connecting to your VPN, such as Proton VPN, you conduct the same search. Does your ISP know you conducted a search on Google? No. Does your VPN provider know you conducted a search on Google? Yes. Does your VPN provider know you searched for "inteltechniques"? No. Why does this matter?

Everyone has a unique threat model, but I will present a few scenarios where you may be concerned. First, consider that I am suing you through civil court, and I have convinced a judge to grant me a court order to collect your internet activity. Since I know where you live, I can assume the provider of your internet service. A court order is issued to your ISP for your internet activity. If your ISP logs your traffic, which most do, the response would tell me every domain which you visited and the dates and times of occurrence. I could use this to prove you were visiting specific websites or transmitting large amounts of data to designated services. If you had a VPN enabled, I could only prove your device(s) were connected through a VPN. I would not know any domains from your activity. A second court order to the VPN provider would not reveal this data. Reputable VPNs do not log this traffic, and IP addresses are shared between thousands of users.

Next, assume I want to know where you live. I know your email provider is Gmail, and a subpoena to them would reveal your IP address at a specific date and time. If this IP address belongs to your internet service provider, a second subpoena will disclose the address of service (your home). If the IP address belongs to your VPN provider, it will not disclose any details about you or the VPN account. A subpoena to the VPN provider for information about the IP address will reveal no logs and an education about IP address sharing between thousands of strangers.

Now, let's combine the strategies mentioned previously to thwart this behavior. Since you are always connected to a VPN, your ISP knows nothing about your internet traffic. A subpoena to them would not reveal the sites you visit. Since Proton Mail does not log your IP addresses in clear text, they cannot determine your true IP address. Since Proton VPN and Proton Mail are Swiss-based companies, they would not respond to a subpoena from the United States. If you purchased a VPN service without providing your true name, there is nothing to glean from the VPN provider about your account (such as a personal credit card number or home address). I hope you now see that all of these strategies strengthen each other.

No VPN company is perfect and all expose a potential digital trail. I choose the option which is most likely to protect me because it has the most to lose. If Proton VPN were caught storing or selling user data, their entire company would lose all credibility and many customers. If a company which owns several VPN brands gets caught doing this, they can simply shut one down and spin up a new marketing campaign for another. I believe Proton VPN has more motive to protect their product and reputation than the larger VPN companies.

DNS Configuration

In the simplest explanation, the Domain Name System (DNS) translates domain names, such as inteltechniques.com, into IP addresses in order to locate the appropriate content. In a typical home setup, your internet service provider (ISP) conducts your DNS queries. In other words, your ISP knows every website domain you visit, regardless of SSL encryption, and knows your billing address. If you did not purchase internet service anonymously, then they also know YOU. 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. Whether you use no VPN whatsoever (poor), rely on an application-based VPN directly on a computer (better), or execute a full home firewall (best), you should modify your DNS settings.

It is important to note that if you are using a VPN application on your computer, it will likely ignore any DNS modifications made on your device and use its own server. This is acceptable for many situations, but not ideal for everyone. 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. Overall, the order of DNS usage is as follows.

- If your web browser has a custom DNS assigned, then all queries from within that browser will use the specified DNS, regardless of any other settings within macOS.
- If you have no DNS specified within the browser, your queries will rely on the DNS provided within any running VPN applications on the machine.
- If you have no DNS assigned within the browser, and no VPN service running, then your DNS queries will be conducted based on the provider which is assigned within macOS.
- If you did not assign any DNS provider within the browser or macOS, and are not using a VPN, then your DNS will rely on your network (likely the ISP).

Let's fix all of this! I recommend NextDNS DNS service for most users. Some people prefer niche privacy-focused community-driven DNS providers for their entire household, but I believe these can make our connections stick out more than widely-used secure options.

If you have read my mobile devices guide, then you know I insist on NextDNS as my DNS provider for my GrapheneOS device. This service allows custom domain filtering outside of typical DNS lookup services. I still highly encourage the use of filtered NextDNS on mobile devices. However, I believe the filtering aspect is unnecessary for macOS. If you are using a software firewall to block undesired program connections, and u-Block Origin to filter browsing data, then I believe you do not need NextDNS's desktop filtering features. Since you only need the DNS lookup utility, I believe NextDNS's public servers are the better option for your macOS device. Finally, let's modify our DNS. First, take a look at your Firefox browser and consider the following.

- Navigate back to the "Settings" menu and select "Privacy & Security".
- Scroll down to the "DNS over HTTPS" section.
- Click the "Max Protection" option and select "Nextcloud".

Your browser will now conduct all DNS queries within an encrypted connection to NextDNS, regardless of any other settings within macOS. Note that this only applies to websites visited from within this installation of Firefox, and not to any other applications. A test at test.nextdns.io should display a confirmation that "DOH" (DNS Over HTTPS) is your DNS query protocol. A visit to crypto.cloudflare.com/cdn-cgi/trace should confirm that you SNI is encrypted, and tls-ech.dev should confirm you are using secure ECH. The following steps make NextDNS the default DNS service for the rest of your macOS device.

IMPORTANT NOTE: The following steps should NOT be applied if you have followed my latest guide to create a home network firewall with VPN. If you have set NextDNS as your DNS provider within pfSense, that configuration (DNS over TLS) should be used instead of adding NextDNS to macOS (DNS over UDP). The first option is more secure and encrypted. In other words, only make the following DNS changes to macOS if you are not (or will not be) using my pfSense firewall settings within the *Extreme Privacy: VPNs & Firewalls* digital guide.

- Launch "System Preferences".
- Click "Network" and select your connection.
- Click "Details" then click "DNS".
- Remove any entries with the "-" button.
- Add 45.90.28.0 and 45.90.30.0 as your desired servers and click "OK".

From now on, all domain name queries from your operating system will be conducted by NextDNS. However, these queries are not encrypted! While you can install software which allows encrypted DNS from macOS, this would break Little Snitch. I value Little Snitch's functionality more than encrypted DNS from the OS, and all browser queries are still encrypted via Firefox. This is why a home firewall is so important.

Also, a VPN application may continue to use its own server. If you want to change this you would need to change the DNS provider within your VPN's settings. This will vary by provider, and is not as vital as the previous settings. At the time of this writing, the native Proton VPN macOS application does not allow assignment of a custom DNS server, but PIA does. However, regardless of your VPN application's DNS choice, your Firefox browser will continue to use whatever DNS you specified during the previous steps.

Personally, I use a network-wide home firewall as explained in the ***Extreme Privacy: VPNs & Firewalls*** digital guide. This allows me VPN protection on my machine without a macOS VPN application, and encrypted network-wide NextDNS coverage. For most, the overall goal is to prevent your ISP from snooping on your traffic. All of these situations prevent that. When I am travelling without a firewall, I rely on the Proton VPN app (or any other reliable VPN) to protect my traffic and encrypt my DNS from my operating system. Again, the browser is always encrypted.

Next, you should ensure that your browser connections are actually encrypted. Within Firefox, navigate to https://test.nextdns.io to conduct a test. You should see "protocol:DOH" (DNS over HTTPS). If you see this, you are hiding much of your internet traffic from your ISP and your VPN.

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 use a VPN application on your device, it supplies its own DNS, which prevents your ISP from seeing your history. When you configure NextDNS on your macOS device, all applications default to it, unless a VPN application prevents this. When you configure NextDNS within your browser, all of your DNS traffic is facilitated by them regardless of your ISP, VPN, or macOS settings. When you use a home firewall, it provides backup DNS services across your entire network.

Your web browser is your window to the internet. Please make sure you have hardened it to a level appropriate for your needs, but not to the point which you have restricted your necessary online activity. Having a hardened version of Firefox will provide great privacy from daily invasions into your online behavior, while the stock version of Safari should help keep your most sensitive accounts in good standing.

CHAPTER SEVEN
VoIP SERVICE

Now that you have your computer configured as privately and securely as possible, you may want to use it for traditional telephone calls over your internet connection. As explained in my previous digital guide about mobile devices, 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 plans. Within GrapheneOS (mobile), I relied on an application called Sipnetic and various Voice over Internet Protocol (VoIP) providers for all telephone calls. Desktop macOS systems will rely on an application called Linphone for use with these same providers. 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 these concerns. 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 at any time. It is probable that you will encounter slight variations compared to my tutorial during configuration. Focus on the overall methods instead of exact steps. Please read the entire chapter before making any decisions.

Important Update: 2024 VoIP Landscape

The content within this chapter was originally written and tweaked throughout 2022 and 2023. During that time, I relied solely on Twilio and Telnyx to provide VoIP service for myself and my clients. Today, I prefer VoIP.ms for many reasons which will be detailed later. The entire VoIP.ms section was overhauled in early 2024 in order to present many new desired features. Before proceeding, I want to provide a summary of concerns about the providers.

Twilio: I have heard from many readers that Twilio is now refusing new service to individuals and small companies. Many people are simply unable to obtain new service. Furthermore, in early 2024, my company's Twilio account was suspended for unknown reasons, even though I had a hefty balance and minimal usage. Twilio refused to provide any details and customer support stopped responding to my emails. Finally, Twilio's configuration can be very difficult at times, but stable once established. Twilio also now prevents outgoing SMS messages unless you enroll (and pay) for 10DLC registration (which I do not recommend). I believe Twilio is now the worst VoIP option, but those who have an established account should keep it.

Telnyx: : I have also heard from many readers that Telnyx is now scrutinizing service to individuals and small companies, with many people unable to establish new service. In late 2023, my Telnyx account was suspended for unknown reasons and I had to fight for several days to regain access. Telnyx's configuration can be very difficult at times, but stable once established. Telnyx prevents outgoing SMS messages unless you enroll (and pay) for 10DLC registration (which I do not recommend).

VoIP.ms: A few years ago, I could not establish new service at VoIP.ms without uploading ID and confirming my identity. Today, their new account creation algorithm is less scrutinous, and many people are reporting the ability to open an account easily. VoIP.ms does not refuse service to individuals like Twilio and Telnyx does. **Currently, I believe VoIP.ms is the best VoIP provider for our needs**. Later, we will easily establish new numbers, calling services, text messaging, voicemail, caller ID, and other features without the need to access their API. VoIP.ms does NOT restrict outgoing SMS from individuals and does not require 10DLC registration.

Instead of re-arranging the instructions for Twilio, Telnyx, and VoIP.ms to reflect my new recommendations, I have kept the same order. Moving sections around is unnecessary and could create confusion. I will begin with Twilio as I have in my pervious guides and then explain Telnyx. After getting through those, I will provide an extremely detailed full configuration of VoIP.ms, which I believe is the absolute best route to go. By the end of this chapter, I think you will agree. If you are able to obtain service at all three providers, that is great redundancy. If you only want an easy solution targeted toward individuals, you may want to skip a lot of the nonsense and go straight to the VoIP.ms section.

If you do want to obtain service through Twilio and/or Telnyx, 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 macOS 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 macOS 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 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 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

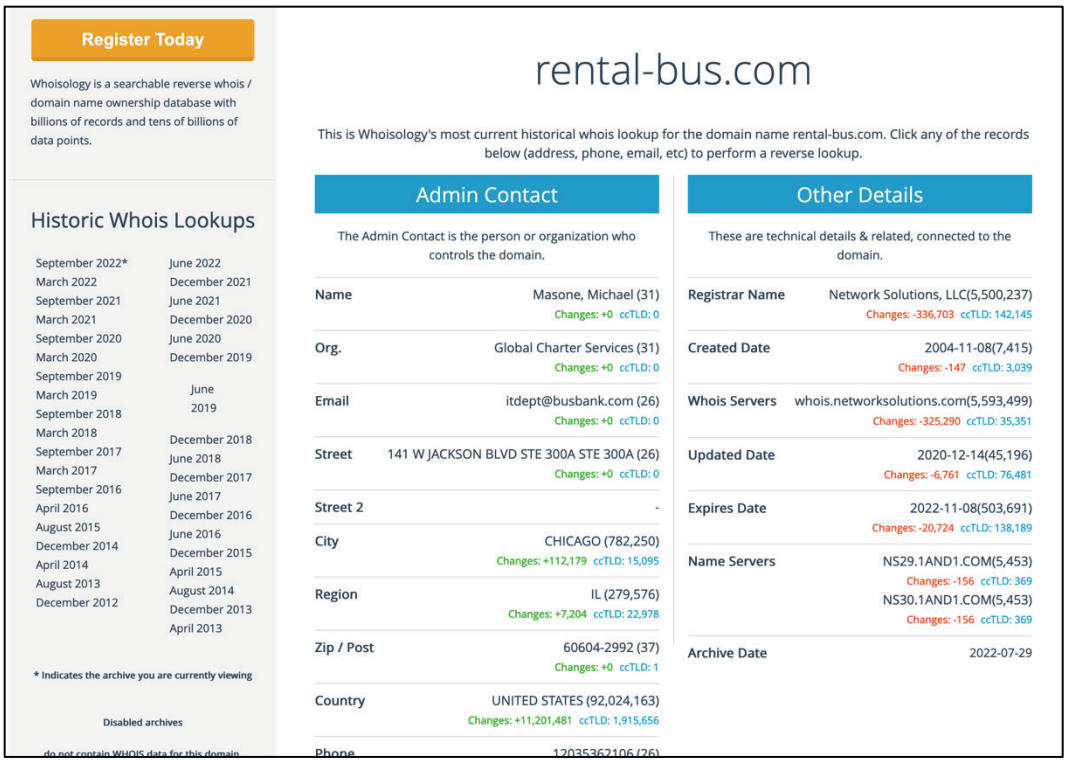
Below is an example of a few random results. While two of these have some internet history, none of them look like a traditional business domain which would pass human scrutiny. I only acquire ".com" addresses for this purpose, as some providers block newer TLDs such as ".work".

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 computer, but I believe it is justified. After purchase, I reserve email addresses associated with this domain solely for use with my new macOS device. When we get into VoIP providers, you will be glad you were proactive with this.

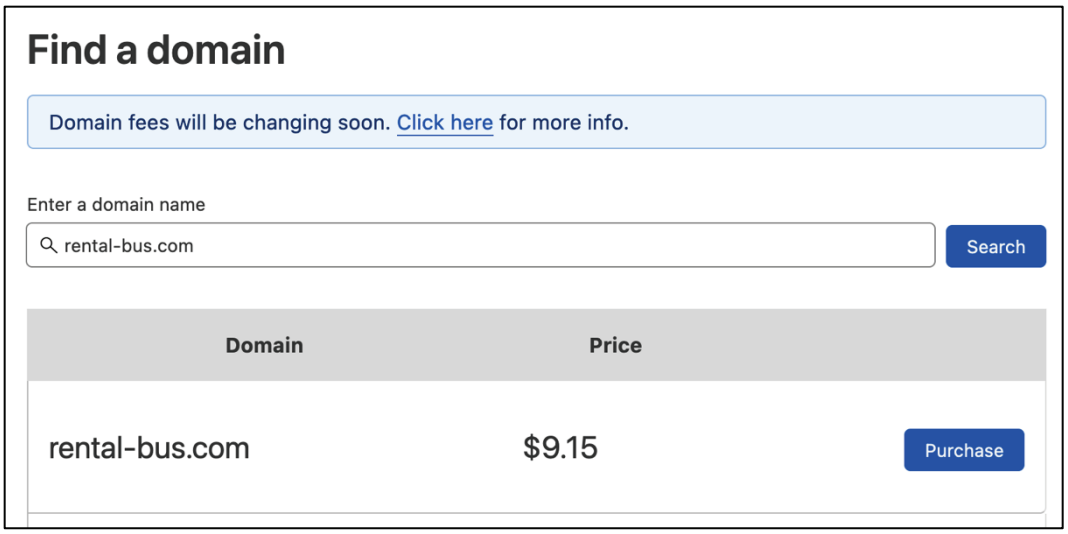
PrairieBoard.com	0	0	-	0				0	1	2 days	available
neamemories.com	0	0	-	0				0	0	Yesterday 19:44	available
FishyChat.com	0	0	-	0				0	0	Yesterday 19:45	available
BankerSkit.com	0	0	-	0				0	0	Today 19:04	available
Rental-Bus.com	0	0	-	0				2	2	3 days	available
biaidi.com	0	0	-	0				0	2	7 days	available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



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.



I purchased a domain for $9.15 and used a masked Privacy.com card for the transaction, but a traditional credit card could also be used. 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 (I have never been asked).

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 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 entirely. 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.

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. 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 Configuration", 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 Linphone Configuration

You now have a SIP domain and credentials created which allow you to associate your Twilio account with VoIP software called **Linphone** (linphone.org). Download it from within Terminal with the following command.

```
brew install --cask linphone
```

The following configuration steps should apply to all Linphone applications, but you may see minor variations across platforms. You will need to repeat each step on every macOS device which you want to use for VoIP calling. Launch Linphone and conduct the following.

- If prompted, click "OK" to allow microphone access.
- If prompted, click "Don't Allow" for camera access.
- If prompted, click "OK" to allow System Events access.
- If prompted by your firewall, allow permanent access to any connection.
- If prompted, accept the program's terms of use.
- If prompted, choose "Use a SIP Account". If this is not present, click the "Home" button and choose "Account Assistant".
- If prompted, click "I understand" about any restrictions.
- Enter a "Username" of your number, such as "2025551212".
- Enter a "Display Name" of your telephone number, such as "2025551212".
- Enter the appropriate "SIP Domain", such as 2025551212.sip.us1.twilio.com.
- Enter the "Password" you previously created for the credential account.
- Change the "Transport" to "TLS". If this ever fails, try "UDP" or "TCP".

Click the confirmations until you return to the main application. You can now click the number selection area in 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 or grey light next to the account if the connection from Linphone to Twilio is successful. We can now make our first call.

- Confirm that your Twilio account is selected within the Linphone application.
- In the search field at the top, input any known telephone number.
- Click the "phone" button to initiate a call.

You should receive an automated message thanking you for using your demo account. I had to click the three dots, then the "Multimedia parameters" in order to select the appropriate incoming and outgoing sound properties. This confirms that we can place calls to Twilio's servers, but we are far from unlimited usage to real numbers. As long as you receive a confirmed test call message from Twilio, 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 the Twilio portal.

- 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. Linphone can be used to place a call at any time from your macOS device. You can also add as many numbers as you wish by repeating this process.

Incoming calls will "ring" your macOS device as long as the Linphone application is open and your status is "green". 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 macOS device. Windows, Linux, Android, and iOS are also supported through Linphone.
- 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 Linphone.
- This process works well with custom Android ROMs, such as GrapheneOS, as explained in the *Extreme Privacy: Mobile Devices* guide.
- 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. While configuring Twilio within the Linphone application during testing of this strategy, I encountered several devices which presented authentication errors during usage. These usually claim that the Twilio credentials supplied to Linphone have failed and the user is prompted to enter the correct password. Supplying the appropriate password fails. This appears to be an issue with Twilio temporarily blocking access due to too many invalid attempts, incorrect protocol settings, or launching and closing of Linphone from mobile devices too many times within a sixty-minute threshold. Any account restrictions should reset after twenty minutes of inactivity, but the following settings within Linphone should mitigate these issues. Navigate to Preferences > Settings > SIP Accounts > Proxy Accounts and click the pencil icon (Edit) and confirm the following.

- Transport: TLS
- Register: Enabled
- Publish presence information: Enabled
- ICE/AVPF/STUN/TURN: Disabled
- Outbound Proxy: Disabled

Linphone software accepts multiple numbers for incoming and outgoing calls. However, their menu only allows you to place outgoing calls from the most recently added (default) number. You can select the default number for outgoing calls within the upper left number selection menu.

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. In a moment we will purge those logs as often as desired.

The biggest feature of this process is the ability to possess affordable VoIP numbers without a third-party service. 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

Linphone 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 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 and the knowledge of uploading files to it.** 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. You may also be blocked from sending SMS messages unless you enroll (and pay) for their 10DLC registration. If you do want to send SMS from this account, you might consider the following.

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 an 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 Twilio 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 Linphone 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 Linphone 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 Linphone 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 Linphone 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 Customer Profiles

In May of 2024, I was notified that my Twilio account must have an approved "Customer Profile" before I could continue making outbound calls from my Twilio numbers. Many readers reported receiving the same email, while many others said they had no restrictions in place. If you only use Twilio to RECEIVE calls and text messages, then you likely need to take no action. Since Twilio blocks outgoing SMS text messages unless you adopt an expensive 10DLC plan, which I never recommend, outbound messaging is likely disabled anyway. However, I often make outgoing calls from my Twilio numbers, and I cannot lose that functionality. If you will ever need to make outgoing calls through a Twilio number, you should create a customer profile.

While logged into your Twilio dashboard, navigate to "Account" > "Customer Profiles" > "Create Primary Customer Profile". If you choose the "Individual" option, you will be required to upload a photocopy of your government ID, which I never recommend. Instead, choose the "Business" option. You will be required to provide the following details.

Legal Business Name, Business Address, Type of Business
Business Registration Number
Domain and Email at Business Domain
Phone Number

I tested these requirements, and determined that you must provide a valid EIN number issued by the IRS. However, Twilio accepted a Sole Proprietorship EIN registered with a DBA (Doing Business As) name and a CMRA "PMB" address. Creation of this exceeds the scope of this book, but it is completely free, immediate, and requires no LLC registration. We are considering a digital guide on the many benefits of Sole Proprietorships. Surprisingly, Twilio accepted a VoIP telephone number and catch-all email address at a registered domain. The approval process was automated, and I was approved within a minute. I could then continue making outbound calls.

I believe the only verification being conducted during this process is to match the provided EIN with the supplied business name, likely accessing an API with the IRS. If you have any EIN (LLC, Sole Proprietorship, Partnership), then you should be allowed to complete the process.

I find these continuous demands from Twilio to be invasive and annoying. While I continue to possess a valid Twilio account with numbers, I only recommend them if you absolutely need them. I prefer VoIP.ms and Telnyx over Twilio any day.

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 manual ways to remove this data, beginning with stored text messages. An automated option is presented soon.

- 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 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".

Custom Twilio Script

If you use your Twilio numbers often, manually sanitizing the data stored on Twilio's servers is quite time consuming. I possess an automated script which conducts all actions on my behalf, and presents information about my numbers, all without logging into my Twilio account. Let's start with some manual commands. Please note that you should replace every instance of "XX" with your Twilio Account SID, and "ZZ" with your Twilio Auth Token, both of which are visible in your console. Each command should be entered on a single line as one execution.

```
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Calls.json -u "XX:ZZ" | python3 -mjson.tool
```

This displays all voice call records stored in your log files.

```
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Messages.json -u "XX:ZZ" | python3 -mjson.tool
```

This displays all SMS text messages stored within your account.

```
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Recordings.json -u "XX:ZZ" | python3 -
mjson.tool
```

This displays all voicemail recordings stored within your account.

```
wget https://api.twilio.com/2010-04-
01/Accounts/XX/Recordings/[SID].wav
```

If you have an audio voicemail message in your account, this allows you to download it. Make sure you replace [SID] with the value presented with the previous query.

```
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Transcriptions.json -u "XX:ZZ" | python3 -
mjson.tool
```

This displays all voicemail transcriptions stored within your account.

```
curl -X GET
"https://lookups.twilio.com/v2/PhoneNumbers/+1[NUMBER}?Fields=
caller_name%2cline_type_intelligence" -u XX:ZZ | python3 -
mjson.tool
```

This displays the caller ID data associated with a specific number (replace [NUMBER]). This often identifies names associated with cellular and VoIP numbers.

```
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Balance.json -u "XX:ZZ" | python3 -mjson.tool
```

This displays your current Twilio balance.

Once you know the SID of a call, SMS, recording, or transcription, you can delete that log file as follows. We will use this in our sanitization script in a moment.

```
curl -X POST "https://api.twilio.com/2010-04-
01/Accounts/XX/Calls/[SID]" -u "XX:ZZ"
```

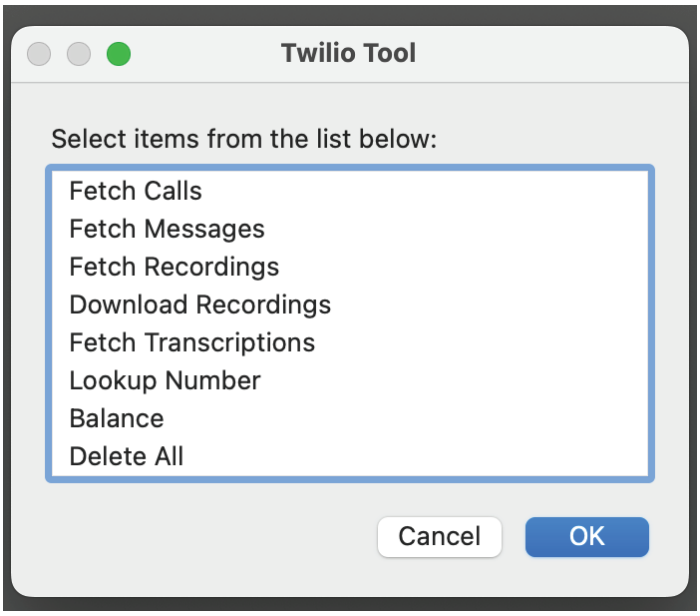
You could memorize these commands and type them when you need them, or create a custom script which would prompt you for data. You could also automate a complete sanitation option to easily wipe out the data stored by Twilio. Let's build it together.

- Open the BBEdit application which we previously installed.
- Copy the bash code within the following pages into BBEdit as new file.
- Change every instance of "XX" to your Twilio account number.
- Change every instance of "ZZ" to your Twilio API token.
- Save the file to your Applications folder as "Twilio" with no file extension.

After it is saved, execute the following within Terminal.

```
cd /Applications && chmod +x Twilio
brew install ncruces/tap/zenity
```

The last command installs a better version of Zenity within macOS for executing bash scripts as graphical menus. You should now be able to double-click the Twilio file from your main Applications menu and launch the script. If it opens within BBEdit, you may need to right-click and select "Get Info", then choose "Terminal" under "Open With". Launching it should appear as follows.



Select any option to see the data associated with the command. The final option is the most powerful. It downloads your call, SMS, recording, and transcription data from Twilio; creates a script for each type of data; extracts the SID numbers from each entry; modifies the scripts to be executable for our needs; deletes the associated data stored by Twilio; and removes the files generated by the script. The script follows.

```
#!/bin/bash

menu=$(zenity --list --title "Twilio Tool" --radiolist --
column "" --column "" FALSE "Fetch Calls" FALSE "Fetch
Messages" FALSE "Fetch Recordings" FALSE "Download Recordings"
FALSE "Fetch Transcriptions" FALSE "Lookup Number" FALSE
"Balance" FALSE "Delete All")

case $menu in

"Fetch Calls")
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Calls.json -u "XX:ZZ" | python3 -mjson.tool
echo ""
echo "Press Enter to Return to Menu"
read data
exec /Applications/Twilio
;;
"Fetch Messages")
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Messages.json -u "XX:ZZ" | python3 -mjson.tool
echo ""
echo "Press Enter to Return to Menu"
read data
exec /Applications/Twilio
;;
"Fetch Recordings")
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Recordings.json -u "XX:ZZ" | python3 -
mjson.tool
echo ""
echo "Press Enter to Return to Menu"
read data
exec /Applications/Twilio
;;
"Download Recordings")
cd ~/Downloads
echo "SID: "
read data
wget https://api.twilio.com/2010-04-
01/Accounts/XX/Recordings/$data.wav
echo ""
echo "Press Enter to Return to Menu"
read data
exec /Applications/Twilio
;;
"Fetch Transcriptions")
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Transcriptions.json -u "XX:ZZ" | python3 -
mjson.tool
echo ""
echo "Press Enter to Return to Menu"
read data
exec /Applications/Twilio
```

```
;;
"Lookup Number")
echo "Number: "
read data1
curl -X GET
"https://lookups.twilio.com/v2/PhoneNumbers/+1'$data1'?Fields=
caller_name%2cline_type_intelligence" -u XX:ZZ | python3 -
mjson.tool
echo ""
echo "Press Enter to Return to Menu"
read data
exec /Applications/Twilio
;;
"Balance")
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Balance.json -u "XX:ZZ" | python3 -mjson.tool
echo ""
echo "Press Enter to Return to Menu"
read data
exec /Applications/Twilio
;;
"Delete All")
cd ~/Desktop
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Calls.json -u "XX:ZZ" > calls.txt
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Messages.json -u "XX:ZZ" > messages.txt
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Recordings.json -u "XX:ZZ" > recordings.txt
curl -G https://api.twilio.com/2010-04-
01/Accounts/XX/Transcriptions.json -u "XX:ZZ" >
transcriptions.txt
grep -o '\"sid\"......................................"'
calls.txt > calls.sh
grep -o '\"sid\"......................................"'
messages.txt > messages.sh
grep -o '\"sid\"......................................"'
recordings.txt > recordings.sh
grep -o '\"sid\"......................................"'
transcriptions.txt > transcriptions.sh
sed -i "s/[\"]//g" *.sh
sed -i "s/sid\: //g" *.sh
sed -i 's/^/curl \-X DELETE
\"https\:\/\/api\.twilio\.com\/2010\-04\-
01\/Accounts\/XX\/Calls\//' calls.sh
sed -i 's/$/\" \-u \"XX\:ZZ\"/' calls.sh
chmod +x calls.sh
./calls.sh
rm calls.txt calls.sh
sed -i 's/^/curl \-X DELETE
\"https\:\/\/api\.twilio\.com\/2010\-04\-
01\/Accounts\/XX\/Messages\//' messages.sh
sed -i 's/$/\" \-u \"XX\:ZZ\"/' messages.sh
chmod +x messages.sh
```

```
./messages.sh
rm messages.txt messages.sh
sed -i 's/^/curl \-X DELETE
\"https\:\/\/api\.twilio\.com\/2010\-04\-
01\/Accounts\/XX\/Recordings\//' recordings.sh
sed -i 's/$/\" \-u \"XX\:ZZ\"/' recordings.sh
chmod +x recordings.sh
./recordings.sh
rm recordings.txt recordings.sh
sed -i 's/^/curl \-X DELETE
\"https\:\/\/api\.twilio\.com\/2010\-04\-
01\/Accounts\/XX\/Transcriptions\//' transcriptions.sh
sed -i 's/$/\" \-u \"XX\:ZZ\"/' transcriptions.sh
chmod +x transcriptions.sh
./transcriptions.sh
rm transcriptions.txt transcriptions.sh
echo ""
echo "Press Enter to Return to Menu"
read data
exec /Applications/Twilio
;;
esac
```

You can now execute the script and select appropriate options. You will be prompted to return to the menu in case you want to conduct a new task. You can press Escape or click cancel to close the application.

This could be used to display your Twilio data or delete the logs of your actions. The effort taken today to customize your own script will save you hours in the future, and allow you to interact with your Twilio account via Terminal without the need to log in to an online portal or rely on a third party. It helps me quickly check messages, calls, and voicemail while ensuring that I am not leaving traces of my activities behind within Twilio's servers.

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 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 some 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 easier. Now that you have an understanding of Twilio, 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 in to 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 Linphone Configuration

- Launch Linphone and click the "Home" button.
- Choose the Account Assistant.
- Choose "Use A SIP Account".
- Enter a "Username" of your login name provided by Telnyx.
- Enter a "Display Name" of your telephone number, such as "2025551212".
- Enter the "SIP Domain" as "sip.telnyx.com".
- Enter the "Password" of your login credential provided by Telnyx.
- Change the "Transport" to "TLS". If this ever fails, try "UDP" or "TCP".

Your Linphone 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", then click "Numbers" > "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 (but not the log).

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**. You will likely encounter a requirement to register your number for a 10DLC campaign which is costly and requires you to upload identification. You may be blocked from outgoing SMS altogether. I only need to receive the occasional SMS from Twilio or Telnyx, which forwards to my email.

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.

Telnyx Voicemail Configuration

Telnyx now offers an official voicemail option, which was recently removed from Beta. It is bare-bones, but it works. Conduct the following to enable the feature.

- Navigate to "Numbers" > "My Numbers" in the Telnyx portal.
- Select the pencil icon next to the desired number.
- Click the "Voice" tab and enable the "Voice Mail' toggle.
- Provide a secure PIN and click "Save".

When anyone calls your number and you do not answer, they will be greeted by a generic voice recording and can leave a voice message. However, you do not get notified when a message has been received. You must call *98 from your Telnyx VoIP number and then conduct the following.

- Enter your PIN and press "#", then press "1" to hear new messages.
- Press "2" to delete any desired messages.

This implementation reminds me of the first days of cellular telephones when you called your own number to find out if you had any messages. Telnyx does offer the ability to slightly customize this greeting, but I found it to be unreliable. I currently do not use the voicemail feature from Telnyx because I do not want to constantly check to see if messages are present, and I do not want messages remaining on their servers.

While the overall Telnyx 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 could be transcribed and emailed to you. With Telnyx, unanswered calls can forward to voicemail, but you will need to call from that number to get your messages. If you want voicemail notifications, files, 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.

If all of this sounds too complicated, then the next service may be more appropriate. VoIP.ms allows all of the features we have previously discussed within their online portal, but everything is easy to enable. Outgoing SMS is allowed, and forwarding incoming SMS is native within their service. They even provide an open-source mobile application to make texting simple.

VoIP.ms Service

I first attempted to obtain VoIP service from VoIP.ms several years ago. Every time I opened an account it was immediately suspended and support staff demanded I send an unredacted photo ID to them. I always refused. This was disappointing because VoIP.ms offers a unique way to obtain two-way SMS messaging along with voice services. In July of 2023, I reached out to the CEO and asked about the issue. He stated that they are currently attempting to modify their fraud controls with hopes of minimizing the need for an ID due to Know Your Customer (KYC) regulatory requirements. Today, I possess a working VoIP.ms account, and I encourage you to test their service. If you are able to secure a functioning account, I think you will find it much easier than the previous two providers. First, you need an account by registering at **https://voip.ms/en/code/IntelTechniques**.

Using this link at signup should provide you $10 in free credits **after you add $15 to your balance**, and a bit less scrutiny on your new account. However, please note all of my comments at the beginning of this chapter. If you supply "John Doe" with a burner email and CMRA address, expect to get suspended. Since we will be using these VoIP numbers with people in our circles as an alternative to our cellular number, I see no reason to use an alias name for this service. I also recommend providing an established personal email account and then updating it to something more secure after account approval. Using a previous physical address may bypass verification.

In early 2024, this affiliate link's stats showed that 40% of the people who used the link were able to open an account while 60% were denied pending identity verification (we do not see any names or other account details). I have no solutions to this but to use your real information. If you are able to obtain service, consider adding a number and connecting it to Linphone with the following steps (instructions to activate VoIP.ms within Sipnetic or Groundwire on mobile systems are in the Mobile guide).

- Log in to your VoIP.ms portal within a web browser.
- Navigate to "DID Numbers" > "Order DID(s)".
- Select your desired country, state, and location, then view numbers.
- Select your desired number and plan (I prefer "Per Minute").
- Choose a server close to you, click "Order DID", and confirm order.
- Click "Sub Accounts" > "Create Sub Account".
- Amend the username with your 10-digit number (12345_2025551212).
- Enter a strong password.
- Select your DID number as the caller ID and click "Create Account".
- Navigate to "DID Numbers" > "Manage DID(s)".
- Select your number and click "Edit Selection-All Settings at Once".
- Change "SIP/IAX" to the new Sub Account and apply changes.
- Open the Linphone app.
- If prompted, click "OK" to allow microphone access.
- If prompted, click "Don't Allow" for camera access.

- If prompted, click "OK" to allow System Events access.
- If prompted by your firewall, allow permanent access to any connection.
- If prompted, choose "Use a SIP Account". If this is not present, click the "Home" button and choose "Account Assistant".
- If prompted, click "I understand" about any restrictions.
- Enter a "Username" of your Sub Account username previously created.
- Enter a "Display Name" of your telephone number, such as "2025551212".
- Enter the "SIP Domain" previously selected, such as "atlanta.voip.ms".
- Enter the "Password" you previously created for the credential account.
- Change the "Transport" to "UDP" (If this fails, try "TCP") and click "Use".

You should now be able to place a test call from this account through Linphone in the same way as the previous tutorials. If you plan to only use the voice features of VoIP.ms, you are all set. However, text messaging is different than the other providers. VoIP.ms is unique in that they offer true two-way SMS text communication. Conduct the following if you want full 2-way SMS functionality within Linphone.

- Navigate to "DID Numbers" > "Manage DID(s)".
- Select your number and click "Edit Selection-All Settings at Once".
- Enable "Message Service" and "Link the SMS received to this DID..."
- Select your Sub Account next to "Link the SMS..." and apply changes.

You should now have the option to send SMS messages from within Linphone, and incoming SMS messages should forward to Linphone while the application is open. You may see messages in the default profile instead of the proxy. If you are you are an Android user, I recommend the official VoIP.ms SMS application available within F-Droid. Apple iOS users should consider Groundwire. I provide full tutorials for each of these options within the ***Extreme Privacy: Mobile Devices*** digital guide.

If you prefer to forward all incoming SMS text messages to an email address **instead** of the VoIP.ms app or Linphone, conduct the following within the VoIP.ms portal.

- Navigate to "DID Numbers" > "Manage DIDs".
- Click the "Edit DID" icon next to your desired number.
- In the "Message Service" section, select the email forwarding option.
- Enter your desired email address and apply changes.
- Deselect the "Link the SMS received to this DID to a SIP Account" option.
- Apply all changes.

Since I use the mobile application for sending and receiving SMS messages through VoIP.ms, I do NOT perform these tasks. You should select only one option. If you prefer mobile access to SMS texting, VoIP.ms with their mobile app is the best option, as explained in ***Extreme Privacy: Mobile Devices***. If you need 2-way SMS messaging on a desktop system, Linphone is the best option. If you only need incoming SMS access without sending messages, email forwarding is the best option.

VoIP.ms Voicemail Configuration

The following configures voicemail access for your account.

- Navigate to "DID Numbers" > "Voicemail".
- Click "Create new voicemail account".
- Enter a Voicemail Number as your telephone number.
- Enter a Name as your telephone number.
- Enter a 4-digit numeric password and enter an email address.
- Change "Delete voicemail message" to "Yes" and click "Create voicemail". If this setting is not present, return to this menu after you save the following modifications and change it.
- Navigate to "DID Numbers" > "Manage DID(s)".
- Select your number and click "Edit Selection-All Settings at Once".
- Change "Voicemail associated with DID" to the mailbox of your number.
- Apply all changes.

Anyone who calls your VoIP.ms number will now be greeted with an option to leave a message. When they do, you will receive an email with an MP3 audio attachment of their message, and the audio will be removed from the VoIP.ms servers. I believe this is the simplest voicemail solution and all providers should make it this easy. With this configuration, there is never a need to enter your voicemail system from a telephone. Everything will work behind the scenes and send voicemails to your email. However, there may be a desire to interact with the system directly. You can call *97 from your VoIP.ms number and you will be prompted to enter your mailbox number and PIN. Once you do, you could record a new greeting or change other mailbox behaviors. I prefer to leave the default settings.

While I like a generic voicemail greeting, you might want a customized option with your own voice (or someone else's). You can record your greeting

- Navigate to "DID Numbers" > "Recordings".
- Provide a name such as "Greeting"
- Click "Upload new recording".
- Click "Browse" to select your MP3 or WAV file.
- Click "Upload file".
- Navigate to "DID Numbers" > "Voicemail".
- Select the "Edit" option next to your mailbox.
- Click "View Advanced Mode".
- Change "Unavailable Message Recording" to your new greeting.
- Click "Save Voicemail".

Account Sanitization

You can manually delete any stored SMS messages or voicemails from the following URLs. If you set the voicemails to be deleted after delivery, you should never see any within the VoIP.ms portal.

https://voip.ms/m/communications.php
https://voip.ms/m/voicemail.php

VoIP.ms API Access

If you use your Voip.ms numbers often for SMS and MMS text messages, you may want to manually sanitize the data stored on their servers. VoIP.ms does not allow you to delete call logs like Twilio, but you can purge your SMS and MMS text messages fairly easily through Terminal. The following two commands display all SMS and MMS text messages stored within your account. Please note that you should replace every instance of "XX" with your VoIP.ms API username, and "ZZ" with your API password, both of which are visible in your VoIP.ms console. I do not include an automated script for this because I have seen different accounts possess unique parameters. If desired, you could adapt the Twilio script for your VoIP.ms needs.

```
curl
'https://voip.ms/api/v1/rest.php?api_username=XX&api_password=
ZZ&method=getSMS'

curl
'https://voip.ms/api/v1/rest.php?api_username=XX&api_password=
ZZ&method=getMMS'
```

Once you receive the ID of a SMS or MMS text, you can delete that log file as follows.

```
curl
'https://voip.ms/api/v1/rest.php?api_username=XX&api_password=
ZZ&method=deleteSMS&id=[ID]'

curl
'https://voip.ms/api/v1/rest.php?api_username=XX&api_password=
ZZ&method=deleteSMS&id=[ID]'
```

Since you can also delete these messages in the VoIP.ms portal, this may be overkill.

Account Creation Roadblocks

If you are asked to provide proof of identity during account creation, I encourage you to resist. Several people have had success telling support they are following this specific guide and trying to "leave Twilio for all of our VoIP needs". I can't guarantee they will waive you in, but they have been very willing to work with our community. You might, send them a link to https://inteltechniques.com/blog/2024/02/27/leaving-twilio-for-voip-ms and tell them you are following that guidance.

Incoming Caller ID

If desired, you can force your calling application to display the name of any incoming caller as it appears within a nationwide CNAM caller ID database.

- Navigate to "DID Numbers" > "Manage DID(s)".
- Select your number and click "Edit Selection-All Settings at Once".
- Enable "CallerID Name Lookup" and apply all changes.

For each incoming call, you will be charged an additional $0.008, but I believe this is worth it. During my testing, I called my new VoIP.ms number from a Google Voice account associated with my name. Linphone displayed "MICHAEL BAZZELL" along with my number, and the voicemail email stated "You have a new message from MICHAEL BAZZELL" followed by my number. For less than a penny, I think this feature is justified.

Encrypted VoIP Communications

Throughout several years, I have received a common piece of feedback about the VoIP content of my books. Several readers have questioned my reasons to promote standard protocols such as UDP and TCP over unencrypted voice when more secure options such as TLS and media encryption were available. There are several reasons, which I will explain, but I will also present steps to add another layer of security.

First, many software VoIP clients have not always played well with these secure options, and some still do not. For many years, missed incoming calls and incomplete outgoing calls were common when encryption was enabled. Things are better today, but not always perfect.

Second, UDP is the standard VoIP protocol which works on almost every provider without any configuration. However, the more secure option of TLS must be explicitly enabled through many providers. This change introduces a slightly larger overhead to the communications, and calls through networks with a weak signal or limited bandwidth can be an issue. As our wireless networks continue to improve and give us stable speed, this is getting better.

Finally, telephone calls should never be considered secure. While we can introduce a secure connection to our VoIP providers via TLS, and encrypt the audio content from our device to their servers, this does not offer true end-to-end encryption. The moment I call you from my VoIP service to your cellular or landline device, there are many hops which must occur, and all of them introduce vulnerabilities for intrusion. Call metadata is left everywhere, and it is likely to be captured by government entities at some point. VoIP calls are intended for non-sensitive tasks, such as calling a business to see if they are open, or making a reservation a restaurant. For these tasks, I do not care much about encryption.

However, there are benefits to encrypted VoIP traffic which justifies experimentation. With some standard protocols, someone sniffing your local network could potentially download the audio from your call. I have demonstrated this in the past at live events when a volunteer in the class made a VoIP call through the same Wi-Fi to which I was connected. I was able to acquire the audio file and play it back through my computer. This is a very targeted attack and required me to be on the same network as the target. In theory, your ISP or VPN provider could attempt the same attack, but I think it would be unlikely.

Nevertheless, let's understand our options, test the strategies with our own devices, and proceed with the most functional options available to us. I will present the steps I took and the outcome of each. Hopefully it will help you decide if VoIP encryption is appropriate for you.

Twilio Encrypted VoIP Communications

Twilio does not offer encrypted communications by default. It must be enabled within the Twilio portal with the following steps.

- Navigate to "Voice" > "Mange" > "SIP Domains".
- Select your configuration and click "Disabled" under "Secure Media" to enable the option.
- Click "Save".

Twilio is now configured to allow encrypted calling, but it is not yet enabled within your software client. The following steps must be taken within Linphone, if you use that product.

- Launch the Linphone "Preferences" or "Settings" menu.
- Click the pencil icon to edit your Twilio SIP settings.
- Change "Transport" to "TLS" and click "Confirm".
- Click "Calls and Chat" and change "Encryption" to "SRTP".
- Enable "Encryption is mandatory" and click "OK".

Generate a test call and ensure that a green shield is visible. You can hover over this to confirm encryption. However, incoming calls will be blocked if you did not complete the previous instruction under Sipnetic and Groundwire. You must navigate to the "incomingvoice" TwiML Bin which you previously created and add ";transport=tls;secure=true" at the end of the Sip entry. Mine appeared similar to the following.

<Sip>2025551212@2025551212.sip.us1.twilio.com;transport=tls;secure=true</Sip>

Test an incoming call and ensure that a green shield is visible. You can tap this to see the details of your call security. Linphone should now be configured to encrypt all incoming and outgoing calls to Twilio's servers. However, I have experienced blocked incoming calls on some Linphone devices. This appears to be more targeted toward

macOS users than Linux or Windows. We are continuing to investigate the reasons Twilio does not send all incoming calls to Linphone on some machines.

Telnyx Encrypted VoIP Communications

Telnyx also does not offer encrypted communications by default. It must be enabled within the Telnyx portal with the following steps.

- Navigate to "Voice" > "SIP Trunking".
- Click the pencil icon to edit your configuration.
- Click "Inbound"; change "Encrypted Media" to "SRTP"; and click "Save".

Telnyx is now configured to allow encrypted calling, but it is not yet enabled within your software client. The following steps must be taken within Linphone, if you use that product.

- Launch the Linphone "Preferences" or "Settings" menu.
- Click the pencil icon to edit your SIP settings.
- Change "Transport" to "TLS" and enable "Publish presence information".
- Change:
  <sip:sip.telnyx.com;transport=tls>

  to

  <sip:sip.telnyx.com;secure=true;transport=tls>
- Click "Confirm".
- Click "Calls and Chat" and change "Encryption" to "SRTP".
- Enable "Encryption is mandatory" and click "OK".

Generate a test call and ensure that a green shield is visible. You can hover over this to confirm encryption. Also generate an incoming test call and ensure that a green shield is visible.

VoIP.ms Encrypted VoIP Communications

VoIP.ms also does not offer encrypted communications by default. It must be enabled within the VoIP.ms portal with the following steps.

- Navigate to "Main Menu" > "Account Settings" > "Advanced".
- Change "Encrypted SIP Traffic" to "Yes" and click "Apply.
- Navigate to "Sub Accounts" > "Manage Sub accounts".
- Click the "Edit Sub Account" icon to the right.
- Change "Encrypted SIP Traffic" to "Yes" and click "Update Account".

VoIP.ms is now configured to allow encrypted calling, but it is not yet enabled within your software client. The following steps must be taken within Linphone, if you use that product.

- Launch the Linphone "Preferences" or "Settings" menu.
- Click the pencil icon to edit your SIP settings.
- Change "Transport" to "TLS" and click "Confirm".
- Click "Calls and Chat" and change "Encryption" to "SRTP".
- Enable "Encryption is mandatory" and click "OK".

Generate a test call and ensure that a green shield is visible. You can hover over this to confirm encryption. Also generate an incoming test call and ensure that a green shield is visible.

Are TLS and secure media configurations justified for your usage? Only you can determine that. I highly recommend that you experiment with these settings and ensure stable calls before you lock them in for full-time use. I currently enable all of these security settings on my own devices, but many clients have had minor issues when relying on them for daily calls. I offer mobile device considerations for Sipnetic and Groundwire in the Mobile Devices guide.

Since traditional telephone calls are never private or secure, I do not heavily push people into these modifications. However, the extremists out there may welcome this small layer of privacy and security. If you encounter missed calls or unstable connections, then the previous non-encrypted options will be better for you. VoIP services are no good to us if we cannot rely on them.

Summary

I currently maintain numbers through all three services and configure each into Sipnetic for mobile and Linphone for desktop. If I were forced to rely on only one service, it would be VoIP.ms due to the simplicity, stability, two-way SMS, voicemail forwarding, and ability to easily sanitize things in my account. If you have a Telnyx or Twilio account and it works for you, great. There is no perfect option for everyone. Today, if a client wants easy access to a VoIP number with turn-key service, I configure a VoIP.ms account for them.

Anticipate fraud-related hurdles from all three providers, but know that you can usually break through the temporary annoyances. Many people ask about services such as JMP.chat. JMP also uses Twilio numbers, but charges $3 monthly (over three times the cost). I see no reason to pay that to a middle man when you could buy your own numbers for much less.

Issues

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 used to send SMS messages with verification codes. I think of these numbers as landline replacements which allow me to send and receive voice calls and personal texts. I maintain a 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 VoIP number(s). This may be desired by some, but could be a disinformation campaign for others. On one of my 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. All three providers require you to contact their support in order to request these modifications, and expect a $10 fee. This may be overkill for most readers.

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 device when needed without exposing my true cellular number. I can accept incoming calls on my devices as if they were traditional landline telephones. The person on the other end does not know I am using VoIP instead of a standard phone line. VoIP calls on a stable internet connection can be more reliable than cellular calls with weak signal.

VoIP Number Decisions

My accounts from Twilio, Telnyx, and VoIP.ms present over a dozen phone 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. However, how many numbers do you really need? This will vary for every reader. Most of my clients only need two VoIP numbers, as explained below.

Personal Number: This is the number you would give to people who know your true name in place of providing your real cellular number. This could include friends and family members who refuse to move over to secure communications. It could be a ported number from your previous cellular account or a brand-new number with a fresh start. I have a VoIP number which is the default communications for people from my past who will never embrace Signal or another secure method of

communication. If I search that number within caller ID services, it displays my full name. It has served its purpose well and kept my true cellular number private.

Alias Number: This is the number you would use in any situation which you do not want associated with your true name. This could be a number to provide to a restaurant while waiting for a table or the mechanic who is working on "John Smith's" vehicle. It is a junk number available to you when you do not want the company seeing your name as the owner when they use caller ID lookup services within their systems.

I believe two numbers aside from your true cellular number are the minimum requirement for our VoIP needs. However, you can take it further if desired. Many of my clients isolate a VoIP number specifically for use with their employer. This prevents co-workers from knowing your personal numbers and allows you to "turn off" when needed without pausing all communications from friends and family. Some clients get addicted and possess over twenty numbers for various purposes.

I always recommend starting small and working your way up. While I enjoy having many numbers at my disposal, I also pay a premium for that luxury. Many of my numbers are never used throughout the month, but I still pay a monthly fee for access. Only add new numbers when you are aware of the specific need for them.

VoIP Fax Service

I have not thought much about Fax machines over the past decade. In the last week, I needed to access a Fax on two separate occasions. The first was to receive a vision prescription. The office said they had no way to email it to me but could fax it. I thought I was stuck in 1999. The second was a request from the IRS. A form from a 2022 filing was missing a single character which identified the category of the submission. The IRS representative needed the entire form filed again, and email was not an option. I could mail it in, but it would never be processed before the due date. She told me Fax was the only option.

I could have used one of several online Fax services, but I trust none of them, especially the free options. My local UPS store has not possessed a Fax machine in several years. My library had one, but it was out of service. I decided to finally create my own Fax service option within my VoIP providers. I doubt this section will apply to many readers, but having a Fax option in our arsenal is a good thing.

Twilio terminated their Fax service in 2021, but both Telnyx and VoIP.ms still offer the option. VoIP.ms is the easiest option and all steps can be completed all within their website. However, it costs twice as much as Telnyx at $2.00 monthly for a number. Telnyx is only $1.00 per month, but you must use their API to send a Fax, and an online web-hook to view anything received. Both options are reliable. If you have a need for constant Fax service, I prefer VoIP.ms over Telnyx. If you only need to send an occasional Fax and want to save a small amount of money, Telnyx works fine. If you live in 2024 and see no need for Fax service, none of this is of you. The following steps assume you already possess one of these VoIP services.

Telnyx Fax

- Navigate to https://portal.telnyx.com/#/app/call-control/fax.
- Click "Create Your First Application".
- Provide a name of "Fax".
- Open a new tab and navigate to https://webhook.site.
- Copy your unique URL, such as: https://webhook.site/975-df-44-8e-d209.
- Paste this URL in the Telnyx tab under "Send a Webhook".
- Click "Save".
- Click "Numbers" > "My Numbers" > "Search and Buy Numbers".
- Choose your country; select "Fax"; and choose an area code.
- Click "Search Numbers" and add a number to your cart.
- Click the Cart and assign this number to your new Fax application.
- Click "Place Order".
- Click "Voice" > "Outbound Voice Profiles".
- Click either "Create your first profile" or "Add new Profile".
- Provide a name of "Outbound Fax" and click "Create".
- Click "Add Connections/Apps to Profile".
- Select "Fax" and click "Add Connections/Apps to Profile".
- Add "North America" or additional countries to outbound calling.
- Click "Save".
- Return to https://portal.telnyx.com/#/app/call-control/fax.
- Click "Fax" and note the "App ID".
- Click "Home" and note your API Key and Public Key.
- Modify the following Terminal command with the PDF URL you want to fax; the "App ID" of your Fax Application; the receiving Fax number; and your outgoing Fax number. Submit when complete.

```
curl -X POST https://api.telnyx.com/v2/faxes \
--data-urlencode
"media_url=https://inteltechniques.com/data/test.pdf" \
--data-urlencode "connection_id=2380691280213574817" \
--data-urlencode "to=+18884732963" \
--data-urlencode "from=+16055020302" \
--header "Authorization: Bearer
KEY0182E5BF41190EDB920994CB0DD1950C_x31IEAy9EAvvcKa6snbXck"
```

You can confirm outgoing transmissions at the previous Webhook URL. You can confirm incoming faxes at the same Webhook address. Simply copy the document URL hosted on an Amazon server and paste the URL into a new tab of your browser to see the incoming Fax.

If desired, disable "Fax" at https://portal.telnyx.com/#/app/call-control/fax when not in use to minimize potential spam faxes. Note that Telnyx does not store incoming Fax PDFs more than 24 hours, so you must download them quickly.

VoIP.ms Fax

- Navigate to https://voip.ms/m/dids.php.
- Select the desired country under "Fax Numbers".
- Select your desired state.
- Click "Order" next to your desired area code and complete the purchase.
- Navigate to https://voip.ms/m/fax/index.php.
- Sending: Click "Send a Fax"; enter the receiving Fax number; Enter your desired sending name; select your Fax number; click browse to select a PDF file; enter a confirmation email; and click "Send".
- Receiving: Click "My Faxes" and select the desired incoming fax document.

Overall, the VoIP.ms Fax options are very similar to their SMS service. They will store your content as long as desired and allow you full interaction within their web portal.

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. 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.

If you have an interest in porting your true cellular number into a VoIP account, or forwarding VoIP calls to other numbers, please download my guide ***Extreme Privacy: Mobile Devices***.

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

CHAPTER EIGHT
VIRTUAL MACHINES

Virtual machines (VMs) virtualize or emulate a particular computer system. They are computer operating systems on top of computer operating systems. Most commonly, a software program is executed within an operating system, and individual operating systems can launch within that program. Each virtual machine is independent from the other and the host operating system. The environment of one virtual machine has no impact on any others.

Quite simply, it is a way to have numerous computers within your single computer. VMs offer a "clean" environment with no contamination from other internet usage. You will be able to clone an original VM in minutes and trash them immediately. We will use virtual machines in order to isolate our sensitive computer usage from the daily driver which gets bombarded with online tracking. The following outlines only a few uses for virtual machines.

Banking: You can keep a VM designated for anything associated with financial transactions. This includes online bill pay, employee payroll, and investment accounts. This way, you know that the VM is free of any viruses or malicious applications. Since it is never used outside of banking, online tracking is minimal.

Shopping: You might rely on various retail outlets for many things. You could boot into a VM designated for online shopping. This VM is never used with any email, social networks, or banking accounts. Furthermore, the entire VM is never associated to your true name. It is only used for ordering items with an alias. This way, you know that Amazon never learns your name or identifies any online browsing history.

Research: You might conduct a lot of investigations. In my book ***Open Source Intelligence Techniques***, I explain how I rely on numerous VMs. Every time I need to research something or someone, I clone my Original VM and open the clone. When finished, I either destroy the clone or export it for archiving. This way, each investigation possesses no contamination from other research.

Sensitive Consultations: When a client needs extreme privacy, I always communicate through a Linux VM which has never been used anywhere else. This is likely overkill, but I justify the paranoia. When communicating through Wire via text through this VM, I know there are no malicious programs, cookies, or other invasive software compromising the communication.

Software testing: Any time I install an application for the first time, I do so within a VM, especially new macOS applications. This allows me to understand the changes the program has made to the operating system, with an easy way to undo any damage. Once I am satisfied that nothing malicious is happening, I proceed to install the software on my main device.

If you want to go the extra mile in achieving extreme privacy, I encourage you to understand virtual machine usage. At any given time, I have no fewer than five VMs ready for action. While most of my VMs are based on Linux, you can also replicate these steps to create a second macOS system. I confess that most casual-usage readers likely have no real need for numerous VMs, but I believe everyone should have one ready to go in case it is needed.

Before creating a virtual machine, you must possess virtual machine software. There are several free and paid programs which allow you to create and execute virtual machines. Premium options such as VMWare offer a free version, but it is extremely limited in function. Parallels works great on macOS, but is expensive. VirtualBox has always been the free gold standard for virtual machine creation, but it does not play well with newer macOS machines. This leaves us with one optimal option, which is **UTM** (mac.getutm.app). I believe this phenomenal free and open-source software works better with macOS than any other free or paid product. Install it from Terminal with the following command.

```
brew install --cask utm
```

UTM allows macOS users to launch practically any virtual system within host machines which have either Intel or ARM (Apple) processors. This means it will work with any Apple computer, regardless of the hardware. UTM employs Apple's Hypervisor virtualization framework to run ARM operating systems on Apple Silicon at near native speeds. On Intel-based machines, traditional x86/x64 operating systems can be virtualized. In addition, lower performance emulation is available to run x86/x64 on Apple Silicon as well as ARM64 on Intel. This allows us practically any option desired, and is unique to this program. Even the paid alternatives do not offer all of these features. The software relies on QEMU, which has always been otherwise difficult to configure. I now rely solely on UTM for all VMs on my macOS machine, and I find it to be superior to a Windows or Linux host.

Upon opening UTM, you have the option to "Create a New Virtual Machine". If this is ever not visible, you can replicate the action by clicking "File" > "New" within the program's menu. Choosing this selection presents options which may be new to readers. You can select to either "Virtualize" or "Emulate" your new VM. We should understand the difference.

Virtualize: This process is accomplished with the help of hardware, typically the hypervisor. It virtually shares the hardware resources of a single physical computer into multiple virtual devices by allocating dedicated resources from the host system to the newly created virtual system. This is typically much faster than emulation and the option we will choose for our new VM. If you have an M1, M2, or newer Apple processor, you cannot virtualize x86/x64 operating systems, you can only virtualize ARM-based systems. Similarly, you cannot virtualize ARM-based systems from a x86/x64 machine. When using virtualization, the operating system must be created for the processor present within the device.

Emulate: This process is much more forgiving, but can be slow. It uses software to emulate specific hardware. This means that the VM needs a software interpreter translating its code into the host system's language. This eats up a lot of resources and can make things drag. Since the VM does not run on the host's physical hardware, emulation is slower when compared to virtualization. By contrast, in virtualization, the guest system gets direct access to the host's allocated resources, resulting in better speed. It is great to have this option, but we will not use it within this chapter.

Let's build our first Ubuntu Linux VM within UTM together. Choose the "Virtualize" option and then select "Linux". This brings us to our need to choose the proper path based on our hardware. You must choose the appropriate version of Ubuntu for your processor. If you have an Intel device, you can use the standard Ubuntu ISO available on their website. If you have an M1, M2, or later processor, you need the ARM version of Ubuntu. Below are the current download links for each.

Ubuntu for Intel Devices: https://ubuntu.com/download
Ubuntu for ARM Devices: https://cdimage.ubuntu.com/jammy/daily-live/current/

If choosing the ARM version, make sure you do not accidentally select the AMD version. Let's talk about these versions a bit more before we proceed. The Ubuntu 22.04 Desktop LTS is the version appropriate for all Windows and Linux hosts, and older macOS machines. It is the long-term version with support until 2027. It is the easy choice for Intel-based machines. Apple M1, M2, and newer machines are tricky. They require an ARM version of Ubuntu, but Ubuntu does not currently offer an official ARM LTS release. However, they do support a daily ARM build which began with the official 22.04 LTS release. We could download the LTS Server edition and modify it for our needs, but I find that unnecessary. The ARM Desktop daily build, based on the current LTS version works well for me. However, I would never use the daily versions of releases such as 22.10, 23.04, 23.10, etc. Those tend to have more bugs and issues. If you are reading this before April of 2024, you should download the daily ARM build of 22.04. In 2024, hopefully we will have an official LTS ARM version. If not, use the ARM Daily version of 24.04.

Once you have downloaded the appropriate version of Ubuntu for your system, click the "Browse" button within UTM and select the downloaded ISO file. Leaving the "Use Apple Virtualization" option unchecked is the safe and stable way to go. However, M1, M2, or newer devices may function better in the future with this option selected. Leave it unchecked for now, as you can always build another machine with this option later to compare the performance. It was not functioning with Linux during this writing. Click "Continue".

Choose half of your system's memory and CPU cores. If you had 16 GB of RAM and an eight-core processor, you would change the RAM to "8192" and the CPU Cores to "4". Never leave the cores as "Default", as it can confuse the operating system. Click "Continue" when complete. The size of the drive should be set to the maximum you will ever need. This is not the size of the VM as it grows, it is only the max. I set mine to "100" GB.

I prefer to enable file sharing, as it makes it easier to extract evidence from your investigation onto your host. Browse to your desired shared folder (I chose Downloads) and click "Continue". Finish through the process and click the arrow icon to start your new Linux VM. You are now ready to install Ubuntu with the following steps. Upon initial boot of your new VM within UTM, conduct the following steps.

Ubuntu 22.04 LTS Desktop Only:

- Select "Try or Install Ubuntu".
- Select "Install Ubuntu"
- Select your desired language and location, then click "Continue".

Ubuntu 22.04 ARM Daily Desktop Only:

- Select "Try or Install Ubuntu".
- Enter the username of "ubuntu" and password of "password" if prompted.
- Double-click "Install Ubuntu 22.04.1 LTS" on the Desktop.
- Choose your language and keyboard layout and click "Continue".

All Ubuntu Installations:

- Select "Normal Installation", "Download Updates", and "Install third …".
- Click "Continue".
- Select "Erase disk and install Ubuntu", then "Install Now".
- Confirm with "Continue".
- Choose your desired time zone and click "Continue".
- Enter a desired name, username, computer name, and password for each field.
- Choose "Log in automatically" and click "Continue".
- Allow Ubuntu to complete the installation and choose "Restart Now".
- Click the "CD" icon in the upper-right of the UTM VM window; highlight the CD/DVD option, and click "Eject". THEN press "Enter" to restart the VM or the "left triangle" within UTM.
- If necessary, choose the "power" or "restart" icons in the UTM window.

Your device should now boot to the login screen. If it boots into the installation ISO again, shut the machine down. In the main UTM window, select your new machine and click the settings icon in the upper-right. Under "Drives", identify the CD/DVD drive and change "Image Type" to "None". Reboot the VM and boot into the Ubuntu Desktop. The following will finish the default configuration.

- Click "Skip" then "Next".
- Select "No" and then "Next" when asked to help improve Ubuntu.
- Click "Next" then "Done" to remove the welcome screen.
- If prompted to install updates, click "Remind me later".
- Continue to the section titled "Ubuntu Customization".

Ubuntu Customization

You should now have an Ubuntu virtual machine executed within your Intel or ARM macOS device. Unlike the steps with VirtualBox, UTM does not require any special extension packs or software to be installed within the VM for the display to resize. You should be able to adjust your screen as desired, or make it full screen for best view. However, do not enter full-screen mode just yet. We still have some work to do and I want you to have easy access to the menu bar. Remember to always use the highest resolution available within Ubuntu and then choose the 100%, 200%, or additional options present within the Ubuntu display preferences. You can right-click the Ubuntu desktop and choose "Display Settings" to see this menu. Also, there are no concerning license restrictions. This is a much better scenario for all of us.

Some desired capabilities, such as clipboard and file sharing, require the installation of UTM's Spice daemon. Launch Terminal from the Applications menu within Ubuntu, and execute the following, which will also modify our background to match the previous tutorial. You may already have the required software, but let's make sure.

- sudo apt install spice-vdagent spice-webdavd
- gsettings set org.gnome.desktop.background picture-uri ''
- gsettings set org.gnome.desktop.background primary-color 'rgb(66, 81, 100)'

Copy and paste capabilities should now be working, but you may not see your shared folder within Files on Ubuntu. The following should fix this.

- Shut down the VM and close the window.
- Within the main UTM window, click the "Settings" icon in the upper-right.
- Select "Sharing" on the left.
- Change "Directory Share Mode" to "Spice WebDAV" and click "Save".
- Reboot the VM.
- Click the "Shared folder" icon in the upper right of the UTM Ubuntu VM.
- Confirm your desired shared folder location.
- Open the Files application within Ubuntu.
- Click the "Other Locations" option in the left menu.
- You should see a folder titled "Spice client".
- Single click that folder and wait for your system to recognize the share.
- Confirm you can access the shared folder within the left menu of Files.

It can take several minutes for the share to become available to Ubuntu. Once it does, it should appear until the VM reboots. Whenever you need to access the shared folder, simply click the Spice folder under "Other Locations" for that session. You can now copy files from your VM directly to your host and vice versa. I typically only do this after the VM has been booted for a while and I need the shared folder. Next, we should consider privacy and security settings within Ubuntu. The first two Terminal commands disable Ubuntu's crash reporting and usage statistics while the remaining steps within Ubuntu's operating system harden our overall privacy and security.

- sudo apt purge -y apport apport-symptoms popularity-contest ubuntu-report whoopsie
- sudo apt autoremove -y
- Launch "Settings" from the Applications Menu.
- Click "Notifications" and disable both options.
- Click the "Privacy" option, then click "Screen" and disable all options.
- Click "File History & Trash", then disable any options.
- Click "Diagnostics", then change to "Never".
- Click the back arrow and click "Power", changing "Blank Screen" to "Never".
- Click "Automatic Suspend" and disable the feature.
- Close all Settings windows.

It is important to keep the software on this original VM updated. There are different ways to do this, but I will focus on the easiest way within the operating system applications. While we do this, it may be a good time to add some commonly used applications to our Dock. Conduct the following steps.

- Launch the Applications menu (nine dots in lower-left).
- Type Terminal into the search field.
- Right-click on the application and select "Add to Favorites".
- Type Software into the search field and right-click on "Software Updater".
- Select "Add to Favorites".
- Press escape until all windows are gone.
- Launch the Software Updater icon from the Dock; click "Install Now"; and update all options.

USB Connection

UTM offers simple USB device access. While any USB drive is attached to the host computer, select the "USB Devices" icon in the upper-right of the UTM VM window. This presents that drive within the dock of Ubuntu for file transfer. If you plug in a USB device while the VM is running, it should prompt you to choose where you want it used.

VM Exports and Clones

While a VM is shut down and the VM window is closed, you will see options within the main UTM application for the selected VM in the upper-right. We will focus on the "Clone selected VM" and "Share" menu options. Clicking the clone option simply presents an option to make a full clone. This allows us to preserve our original VM and make copies whenever we want. The share option allows us to export an entire VM within a ".utm" file for easy archiving or sharing. The dialogue will prompt you to choose the export location. If you ever want to see the location of all VMs within UTM, navigate to the following directory within the macOS host.

/Users/[your macOS username]/Library/Containers/UTM/Data/Documents/

If this location is not convenient, and I believe it is not, you can move your VM to any folder desired with the "Move selected VM" option within the main windows. I keep all of mine within a dedicated folder on my macOS host. Note that you can only move them once using the button option.

Virtual Machine Size & Shrinking

Similar to any other VM software, your VMs within UTM will keep growing unnecessarily. Fortunately, UTM makes the shrinking process easy. While a VM is shut down, select it within the main UTM window. Go to the Settings for that VM and select the IDE drive of your operating system within the left menu. Click the "Reclaim Space" button and confirm the option. You will see the size decrease if free space was available.

macOS Virtual Machines

If you own an Apple computer with an M1, M2, or later Apple processor, you can easily run multiple macOS virtual machines within UTM. This provides a "test" copy of macOS for any experimentation without modifying your host operating system. The following applies only to Apple processors. First, you must download the appropriate "IPSW" file from a legitimate Apple source. I prefer to visit the following website and download the latest "Final" IPSW file, which at the time of this writing was UniversalMac_13.3.1_22E261_Restore.ipsw.

https://mrmacintosh.com/apple-silicon-m1-full-macos-restore-ipsw-firmware-files-database/

The full source of the file was as follows.

https://updates.cdn-apple.com/2023WinterFCS/fullrestores/032-66602/418BC37A-FCD9-400A-B4FA-022A19576CD4/UniversalMac_13.3.1_22E261_Restore.ipsw

Once you have the IPSW file on your machine, you are ready to continue with installation.

- Launch UTM and click "Create a New Virtual Machine".
- Choose "Virtualize" and then "macOS 12+".
- Click "Browse" and select your downloaded IPSW file.
- Modify the RAM to 50 % of your current resources.
- Modify the CPU to half of the "Performance" cores. My 10-core M1 Pro has 8 performance cores, so I chose "4" for this option.
- Click "Continue" and specify the desired size of the drive.
- Click "Continue" then "Save".
- Click the "Settings" icon and click the "Display" menu.

- Choose the display of your device, enabling "HiDPI".
- Click "Save" and exit the settings if necessary.

If you want a shared drive for accessing files across each OS, complete the following.

- From the macOS host, open "System Settings" > "General" > "Sharing".
- Enable "File Sharing" and click the information button.
- Click the "+" and browse to the desired shared folder.
- Click "Add" then "Done".
- In the UTM program, open the settings for the macOS VM.
- Ensure "Network" is set to "Shared Network" and click "Save".
- In the UTM VM settings, click the shared directory menu and select "Browse".
- Select the same folder previously shared and click "Open".
- Launch the macOS VM and open Finder.
- Go to "Finder" > "Settings" > "Sidebar" and enable "Connected Servers".
- Click "Network" within Finder and double-click the host.
- Click "Connect As…" and enter your host OS credentials.
- Enable "Remember this password…" and click "Connect".

If you want to temporarily disable internet connectivity from your host device while you configure your operating system and firewall, conduct the following before the initial boot.

- Right-Click the "Network" menu item and select "Remove".

AFTER you have configured the operating system and firewall within the VM, as explained within chapters three and four, conduct the following to re-enable internet connectivity from the host machine.

- Select the desired VM within UTM.
- Click the "Settings" icon in the upper-right.
- Click "New" in the left menu and select "Network, then click "Save".

Before launching, I recommend modification to the screen resolution. I right-clicked on my machine within UTM and selected the Display menu item. I then changed the resolution to 3840 x 2160 and activated "HiDPI". This substantially increased the VM resolution for my external monitor. You should play with these options until you find the setting appropriate for your machine.

Launch your new macOS VM. You may need to confirm the installation by pressing "OK". Allow the process to complete, which may take some time. Once complete, you have a fully-functioning macOS VM which should be as smooth and responsive as the native host. Rely on the same tutorials within Chapter Three if you want to harden the operating system, or you may want a stock system without any changes. You have the ability to copy files to and from the VM when needed. It could be used

to test software before committing on your macOS host or to conduct sensitive activity which may be inappropriate for your daily host. I conduct many online investigations within both macOS and Linux virtual machines.

Note that the drive for your macOS VM will be 100% full. Whatever size you set will create a file that size for the VM. There is currently no shrinking option. UTM can also host Windows VMs, but they must be based on ARM builds if you want them to load in M1 or newer processors.

Network Connectivity

It is important to note that UTM uses Apple's native virtualization framework on Apple Silicon hardware. Therefore, connections within any UTM virtual machine cannot be intercepted by Little Snitch. This is why you never see options to block traffic from within UTM VMs. I do not have an issue with this because I am mostly running Linux VMs within UTM which do not possess abusive telemetry. However, you may encounter a scenario where you want to run a VM without any network connectivity to the internet. Before launching a UTM VM, conduct the following.

- Select the desired VM and click the "Settings" icon.
- Right-click "Network" in the left menu, select "Remove" then click "Save".

The chosen VM will now possess no internet connectivity and can be safely used offline. To reverse this process, exit the VM and conduct the following.

- Select the desired VM and click the "Settings" icon.
- Click "New" in the left menu, select "Network", and click "Save".

Summary

I currently rely on a 2021 MacBook Pro (M1) for all of my online investigations. I only use virtual machines within UTM and possess no other virtualization software on my machine. I simply find UTM to be much more stable than the equivalents for other operating systems. I can quickly launch multiple Linux, Windows, and macOS VMs for my OSINT needs. A Linux VM boots in less than 10 seconds; a Windows VM in less than 19 seconds; and macOS VMs somehow boot in only 6 seconds. I can easily test software and customizations without much of a time commitment. Using UTM's "Run without saving changes" option allows me to avoid snapshots when I want a disposable machine. We are spoiled.

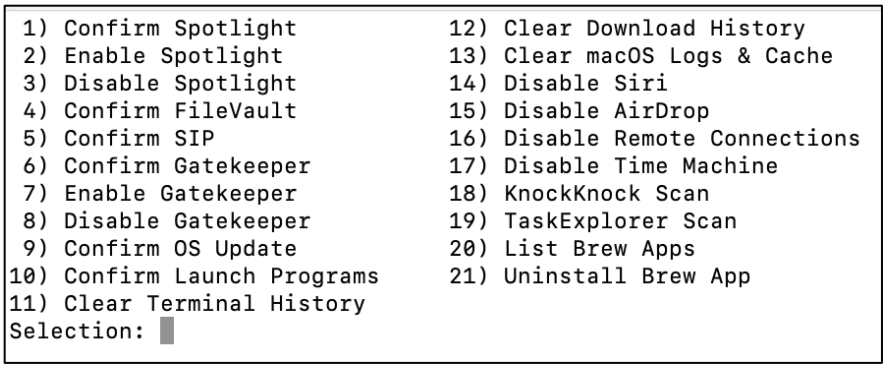
CHAPTER NINE
CUSTOM SCRIPTS

I have explained a lot of ways in which you can harden and sanitize your new macOS machine. Everything up to this chapter relies on manual configuration. Next, let's automate a lot of the daily or weekly tasks by creating our own macOS maintenance script. While I will explain every step, I respect that some readers may just want to download a pre-configured script and use it right away. The following commands within Terminal will download each script and make each executable within your macOS machine by opening them from your Applications folder. The rest of the chapter extensively explains each option.

```
cd /Applications
wget https://inteltechniques.com/data/Terminal-Maintenance
wget https://inteltechniques.com/data/Terminal-Search
wget https://inteltechniques.com/data/Terminal-Updates
chmod +x Terminal-Maintenance
chmod +x Terminal-Search
chmod +x Terminal-Updates
```

Now, let's understand how we could have built these each manually, beginning with the Maintenance script. Copy each piece of code presented over the next six pages in `Courier New 11-point font` and paste the text into a new file saved as `Terminal-Maintenance` within your Applications folder using the BBEdit application on your own macOS. I will explain each segment as we go along, which could be helpful if you decide to make your own scripts.

The text on the following page creates the menu which we will use in our script. It identifies the file as a bash script; clears the screen; and presents a menu with 21 selectable menu options. The image below displays the script once it is finished and executed. You would enter the number associated with the feature you want to execute and strike the return key. Striking the return key at any prompt without a number presents the original menu again.



```
#!/bin/bash
clear
PS3='Selection: '
options=(
"Confirm Spotlight"
"Enable Spotlight"
"Disable Spotlight"
"Confirm FileVault"
"Confirm SIP"
"Confirm Gatekeeper"
"Enable Gatekeeper"
"Disable Gatekeeper"
"Confirm OS Update"
"Confirm Launch Programs"
"Clear Terminal History"
"Clear Download History"
"Clear macOS Logs & Cache"
"Disable Siri"
"Disable AirDrop"
"Disable Remote Connections"
"Disable Time Machine"
"KnockKnock Scan"
"TaskExplorer Scan"
"List Brew Apps"
"Uninstall Brew App"
)
select opt in "${options[@]}"
do
case $opt in
```

Next, our script must identify the tasks to perform when a specific menu item is selected. Let's work through each option. The following, which is listed as "1) Confirm Spotlight" within the executed script, displays the status of your Spotlight indexing. I prefer mine to confirm that both indexing and search is disabled.

```
"Confirm Spotlight")
mdutil -s /
;;
```

The next option, which is listed as "2) Enable Spotlight" in the executed script, allows you to enable Spotlight if desired. It also rebuilds the database.

```
"Enable Spotlight")
sudo mdutil -i on /
sudo mdutil -E /
;;
```

The next option, which is listed as "3) Disable Spotlight" in the executed script, allows you to disable Spotlight if desired. It also deletes the database.

```
"Disable Spotlight")
sudo mdutil -i off /
sudo mdutil -E /
;;
```

The next option, which is listed as "4) Confirm FileVault" in the executed script, displays the current status of your encrypted internal drive.

```
"Confirm FileVault")
fdesetup status
;;
```

The next option, which is listed as "5) Confirm SIP" in the executed script, displays the current status of Apple's System Integrity Protection. I have not previously explained this, but it ensures that malicious software cannot modify protected operating system files. This should be enabled.

```
"Confirm SIP")
csrutil status
;;
```

The next option, which is listed as "6) Confirm Gatekeeper" in the executed script, displays the current status of the Gatekeeper service, which queries Apple's servers to authorize any downloaded or updated programs on your system. I prefer mine to be disabled.

```
"Confirm Gatekeeper")
spctl --status
;;
```

The next option, which is listed as "7) Enable Gatekeeper" in the executed script, allows you to enable Gatekeeper in case you change your mind after disabling it.

```
"Enable Gatekeeper")
sudo spctl --master-enable
;;
```

The next option, which is listed as "8) Disable Gatekeeper" in the executed script, allows you to disable Gatekeeper in case you change your mind after enabling it.

```
"Disable Gatekeeper")
sudo spctl --master-disable
;;
```

The next option, which is listed as "9) Confirm OS Update" in the executed script, displays any pending macOS updates. You will still need to update through System Settings, but this is an immediate way to see them. You must choose the "Apple Update" profile within Little Snitch if you executed the firewall strategy previously explained.

```
"Confirm OS Update")
softwareupdate -l
;;
```

The next option, which is listed as "10) Confirm Launch Programs" in the executed script, opens three Finder windows to display known locations where programs are set to automatically launch upon every boot. If you see anything here which you do not want running in the background at all times, you could remove it.

```
"Confirm Launch Programs")
open ~/Library/LaunchAgents/
open /Library/LaunchAgents/
open /Library/LaunchDaemons/
;;
```

The next option, which is listed as "11) Clear Terminal History" in the executed script, eliminates any stored commands within your previous Terminal sessions. This could be used to wipe out any sensitive input or queries.

```
"Clear Terminal History")
rm -f ~/.bash_history
rm -f ~/.zsh_history
;;
```

The next option, which is listed as "12) Clear Download History" in the executed script, eliminates the file which stores the history of files downloaded to your macOS device from the internet.

```
"Clear Download History")
rm ~/Library/Preferences/com.apple.LaunchServices.
QuarantineEventsV2
;;
```

The next option, which is listed as "13) Clear macOS Logs & Cache" in the executed script, runs several commands which attempt to purge known macOS history logs.

```
"Clear macOS Logs & Cache")
sudo rm -rfv /Library/Logs/*
rm -rfv
~/Library/Containers/com.apple.mail/Data/Library/Logs/Mail/*
sudo rm -rfv /var/audit/*
sudo rm -rfv /private/var/audit/*
sudo rm -rfv ~/Library/Logs/*
sudo rm -fv /System/Library/LaunchDaemons/com.apple.periodic-
*.plist
sudo rm -rfv /var/db/receipts/*
sudo rm -vf /Library/Receipts/InstallHistory.plist
sudo rm -rfv /private/var/db/diagnostics/*
sudo rm -rfv /var/db/diagnostics/*
sudo rm -rfv /private/var/db/uuidtext/
sudo rm -rfv /var/db/uuidtext/
sudo rm -rfv /private/var/log/asl/*
```

```
sudo rm -rfv /var/log/asl/*
sudo rm -fv /var/log/asl.log # Legacy ASL (10.4)
sudo rm -fv /var/log/asl.db
sudo rm -fv /var/log/install.log
sudo rm -rfv /var/log/*
sudo rm -rfv /Library/Caches/* &>/dev/null
sudo rm -rfv /System/Library/Caches/* &>/dev/null
sudo rm -rfv ~/Library/Caches/* &>/dev/null
sudo rm -rfv /var/spool/cups/c0*
sudo rm -rfv /var/spool/cups/tmp/*
sudo rm -rfv /var/spool/cups/cache/job.cache*
sudo rm -rfv ~/.Trash/* &>/dev/null
rm -rfv ~/Library/Developer/Xcode/DerivedData/* &>/dev/null
rm -rfv ~/Library/Developer/Xcode/Archives/* &>/dev/null
rm -rfv ~/Library/Developer/Xcode/iOS Device Logs/*
&>/dev/null
sudo dscacheutil -flushcache
sudo killall -HUP mDNSResponder
sudo purge
;;
```

The next option, which is listed as "14) Disable Siri" in the executed script, runs several commands which attempt to disable all known Siri integrations within macOS. While you have likely already disabled these, I have seen them creep back in after major system updates.

```
"Disable Siri")
defaults write com.apple.assistant.support 'Assistant Enabled'
-bool false
defaults write com.apple.assistant.backedup 'Use device
speaker for TTS' -int 3
launchctl disable "user/$UID/com.apple.assistantd"
launchctl disable "gui/$UID/com.apple.assistantd"
sudo launchctl disable 'system/com.apple.assistantd'
launchctl disable "user/$UID/com.apple.Siri.agent"
launchctl disable "gui/$UID/com.apple.Siri.agent"
sudo launchctl disable 'system/com.apple.Siri.agent'
defaults write com.apple.SetupAssistant 'DidSeeSiriSetup' -
bool True
defaults write com.apple.systemuiserver 'NSStatusItem Visible
Siri' 0
defaults write com.apple.Siri 'StatusMenuVisible' -bool false
defaults write com.apple.Siri 'UserHasDeclinedEnable' -bool
true
defaults write com.apple.assistant.support 'Siri Data Sharing
Opt-In Status' -int 2
;;
```

The next option, which is listed as "15) Disable AirDrop" in the executed script, disables Apple's AirDrop service.

```
"Disable AirDrop")
defaults write com.apple.NetworkBrowser DisableAirDrop -bool
true
;;
```

The next option, which is listed as "16) Disable Remote Connections" in the executed script, runs several commands which attempt to disable all known remote connection processes within macOS. If you have no need to share your screen, printer, or drive with other users on your network, this provides another layer of security protection.

```
"Disable Remote Connections")
sudo systemsetup -setremotelogin off
sudo launchctl disable 'system/com.apple.tftpd'
sudo defaults write
/Library/Preferences/com.apple.mDNSResponder.plist
NoMulticastAdvertisements -bool true
sudo launchctl disable system/com.apple.telnetd
cupsctl --no-share-printers
cupsctl --no-remote-any
cupsctl --no-remote-admin
;;
```

The next option, which is listed as "17) Disable Time Machine" in the executed script, disables Apple's Time Machine backup service. If you have never set this up, and rely on the more thorough solution presented in the next chapter, it is best to turn this off.

```
"Disable Time Machine")
sudo tmutil disable
;;
```

The next option, which is listed as "18) KnockKnock Scan" in the executed script, runs the Terminal version of a full scan using the security program KnockKnock. It then parses the results and only displays each line announcing any hit from Virus Total and the immediately following line which identifies the name of the application. One result from my scan appears immediately below. It confirms that no viruses were detected within my executed Little Snitch application.

```
"VT detection" : "0\/74",
"name" : "Little Snitch Agent",
```

```
"KnockKnock Scan")
cd /Applications
./KnockKnock.app/Contents/MacOS/KnockKnock -whosthere -pretty
> ~/Desktop/KnockKnock.txt
rg -aFiNA 1 "VT Detection" ~/Desktop/KnockKnock.txt
rm ~/Desktop/KnockKnock.txt
;;
```

The next option, which is listed as "19) TaskExplorer Scan" in the executed script, is similar to the previous option. It runs the Terminal version of a full scan using the security program TaskExplorer. This requires your system password. It then parses the results and only displays each line announcing any positive hit from Virus Total identifying a potential virus within a running process. If none are found, you will only see the notice displaying "Process complete. If no results, then nothing was identified as suspicious".

```
"TaskExplorer Scan")
cd /Applications
sudo ./TaskExplorer.app/Contents/MacOS/TaskExplorer -pretty -
explore > ~/Desktop/TaskExplorer.txt
sed -i '' 's/VT detection\" \: \"0//g'
~/Desktop/TaskExplorer.txt
rg -aFiN "VT Detection" ~/Desktop/TaskExplorer.txt
echo "Process complete. If no results, then nothing was
identified as suspicious."
rm ~/Desktop/TaskExplorer.txt
;;
```

The next option, which is listed as "20) List Brew Apps" in the executed script, displays all software packages installed by Brew. This is helpful in identifying the exact name of any programs which are no longer wanted, which will be used for the next option.

```
"List Brew Apps")
brew list
;;
```

The next option, which is listed as "21) Uninstall Brew App" in the executed script, allows you to specify the exact name of a software package installed by Brew for complete removal. Using the "--zap" and "--force" switches within the standard uninstall command and ensures that we remove all possible traces of an application.

```
"Uninstall Brew App")
Echo "Enter App Name: "
read data
brew uninstall --cask --zap --force $data
brew cleanup -s
rm -rf "$(brew --cache)"
brew missing
brew autoremove
;;
```

The next commands simply close our script.

```
esac
done
```

Next, I present the "Terminal-Search" script. You would again copy each piece of code presented within this page in `Courier New 11-point font` and paste the text into a new file saved as `Terminal-Search` within your Applications folder using the BBEdit application on your own macOS. This script presents only four options, as seen immediately after the script. The first allows you to search for any full or partial file name throughout your entire internal drive. The second searches for file names only within the Documents folder. The third searches within the content of your files inside the Documents folder, such as the text within a document. The final option identifies any files larger than a specified size, which can be helpful in identifying large files taking up valuable space. This script attempts to replicate the basic search options which may be missing if you disabled Spotlight.

```
#!/bin/bash
COLUMNS=12
PS3='Selection: '
options=("Search File Names (Root)"
"Search File Names (Documents)"
"Search Content (Documents)"
"Search Size")
select opt in "${options[@]}"
do
case $opt in
"Search File Names (Root)")
echo "Enter Term: "
read data
find / -print 2>/dev/null | grep -i $data
;;
"Search File Names (Documents)")
echo "Enter Term: "
read data
find ~/Documents/ | grep -i $data
;;
"Search Content (Documents)")
cd ~/Documents
echo "Enter Term: "
read data
rg -aFiN $data
;;
"Search Size")
cd /
echo "Enter Size in GB: "
read data
sudo find . -size +"$data"G -exec du -h {} \;
;;
esac
done
```

```
1) Search File Names (Root)
2) Search File Names (Documents)
3) Search Content (Documents)
4) Search Size
Selection:
```

The final script is the "Terminal-Updates" file which automates the Brew update and cleanup process previously presented. You would again copy each piece of code presented over the next section in `Courier New 11-point font` and paste the text into a new file saved as `Terminal-Updates` within your Applications folder using the BBEdit application on your own macOS. There is no menu for this script, it simply executes each command which will disable Brew's analytics; update Brew itself; update each application; force an update of any older versions; cleanup any unnecessary cached files; remove the Homebrew cache itself; replenish any missing dependencies; remove software no longer needed by your computer; and check that everything is configured properly.

```
#!/bin/bash
set -x #echo on
brew analytics off
brew update
brew upgrade
brew upgrade --greedy
brew cleanup -s
rm -rf "$(brew --cache)"
brew missing
brew autoremove
brew doctor
```

You should now see all of these scripts within your Applications folder next to your traditional programs. If you are unable to execute them by clicking or double-clicking each, make sure they are set as executable with the following Terminal commands.

```
cd /Applications/
chmod +x Terminal-Maintenance
chmod +x Terminal-Search
chmod +x Terminal-Updates
```

You can apply these overall tutorials presented within this chapter toward any new desired scripts which can automate Terminal commands. In the next chapter, I explain my scheduling of these and other tasks for routine maintenance.

CHAPTER TEN
UPDATES & MAINTENANCE

I hope you now have your ideal macOS device configured specifically for your needs. Next, you need to make sure you keep it that way. I encourage you to adopt a macOS maintenance schedule which makes most sense for you. My routine is to conduct all computer maintenance on Friday afternoons when my digital workweek is over. It just so happens that I am writing this chapter on Friday, May 12, 2023. This makes it easy to document my entire process, which should be quick thanks to our automated scripts. I try to take the following actions once weekly when I am able to shut my computer down without interrupting any pending work.

First, I open the Terminal-Maintenance script and quickly launch options 1, 4, 5, and 6. These confirm that my Spotlight, FileVault, SIP, and Gatekeeper settings are still configured as desired. This takes less than 20 seconds. I then switch my Little Snitch profile to "Apple Update" and launch option 9 within the Maintenance script to identify any pending macOS updates. If there are any updates which need applied, I make a mental note and return to that later. I then execute option 10 to make sure that no undesired programs have embedded themselves into my computer login process. I launch options 11, 12, and 13 to clear my logs and history. I launch 18 and 19 to make sure that my system has no malicious processes running.

Next, I execute the Terminal-Updates script to update all of my Brew programs and utilities. After that finishes, I open System Settings and apply any macOS updates if there were any waiting. If there were updates, I can expect a system reboot, which is allowed immediately following the update process. If there were no macOS updates, I go ahead and reboot my computer after I have closed all applications and saved all open documents. I switch Little Snitch back to "Apple Disabled" after any updates.

A macOS reboot usually fixes any weird bugs I may have noticed, such as slow application response or delayed software execution. Rebooting clears a lot of application cache which can be troublesome, so I make sure to reboot completely at least once per week. If something still seems unusual, I launch Onyx; select the "Maintenance" option; and enable the following toggles.

- Structure of the file system
- Delete APFS snapshots
- System
- Applications
- Internet
- Other

I then click the "Run Tasks" button and allow the process to complete, which will also reboot the system when finished.

Once my computer is completely updated and has reboot, I like to make a full backup. There are numerous paid backup programs for macOS, but all of them rely on embedded file synchronization features already present in the operating system. Apple's own Time Machine is the most used backup solution, but I find it inappropriate for my usage, and I do not want to risk data collection by Apple or iCloud. I always prefer a third-party solution.

Some free backup apps simply copy your data within the "Documents" folder to an external drive. This might copy your personal files, but it would miss a lot of important data. As one example, the hidden "Library" folder within your home directory contains all of your email messages and many other important files. Therefore, these minimal backup options are not sufficient.

Some premium backup programs proudly claim that their backups are full clones and can be booted from external devices. While this was true for older macOS devices, I find this to fail more than function on modern machines. I also simply do not need to ever boot from my backup drive. I only want the data preserved in the event I would ever need to rebuild my machine. These full disk clones also preserve all standard macOS system files, which we can always get from a fresh new build. I find most cloning programs overkill for my needs.

A paid program is not needed. Instead, I rely on a free and open-source program called FreeFileSync, available at freefilesync.org/download.php. They do not offer an official Brew option, so you will need to manually download the installation file from their site. You will need to work your way through the inline ads, and locate the "Download FreeFileSync macOS" link. Once you have the file, double-click it from the Downloads folder and complete the setup process. I accepted all default options. When Little Snitch alerted me to a connection request to the FreeFileSync servers, I selected "Forever", "Any Connection", and "Deny" into the "Effective in all profiles" option. This prevents all connections, but also requires you to manually update the program whenever needed.

Upon opening the application, you will need to configure it for full disk access. I was also presented with another Little Snitch alert, to which I replicated the previous setting. This is already applied to the Little Snitch configurations previously downloaded. I took the following steps to finish the FreeFileSync installation once presented the "Grant Full Disk Access" window.

- Click the "Open Security & Privacy" button.
- Enable the toggle next to "FreeFileSync".
- Confirm system password and modify the setting.
- Allow to "Quit & Reopen".

Your software should now be ready for use. This can be used to synchronize any two folders, but we will only focus on our Home directory. First, we need to properly format our external drive. I highly recommend an external USB SSD, such as the SanDisk Extreme line of USB drives. If you have a 1 TB or smaller internal drive, the

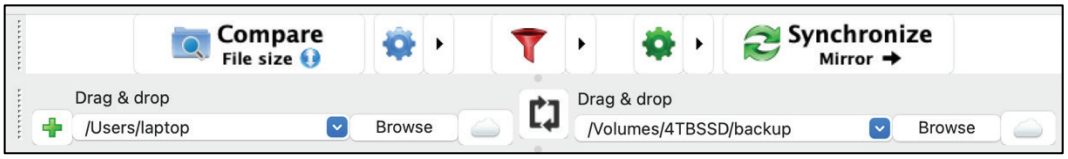
$82 SanDisk 1 TB Extreme Portable SSD (https://amzn.to/42S7x7M) would be perfect. Larger internal drives will require larger external devices. The more expensive "Pro" versions of these drives will not provide much benefit, and are unjustified for our purpose. I format my external SSD specifically for backups with the following steps.

- In macOS, navigate to "Applications" > "Utilities" > "Disk Utility".
- Select the highest listed item for the new drive and click "Erase".
- Name it "Backup" with a Format of "APFS Encrypted".
- Make the Scheme "Guid Partition Map".

Now, let's conduct our first backup within FreeFileSync.

- Click "Browse" in the left "Drag & drop" area.
- Click "Macintosh HD" in the left menu.
- Select "Users" and then your machine's username.
- Click the "Open" button.
- Click "Browse" in the right "Drag & drop" area.
- Select your external hard drive and click "New Folder".
- Create a new folder called "backup" on your external drive and click "Create".
- Click the right arrow icon next to the green cog wheel near "Synchronize".
- Change the option to "Mirror".

The mirror option makes sure that the data on the external drive is always an exact replica of the content on your computer. If your system name was "laptop" and external drive was labeled "4TBSSD", yours might look similar to mine below. You could save this configuration with the "Save as" icon, naming it "Home Backup".



Next, click "Compare" and allow the analysis. You may receive warnings that FreeFileSync is trying to access your Desktop, Download, and other folders, but this is acceptable for this purpose. You may also receive a warning about an area which is inaccessible to the program. I click "Ignore All" when this happens. Once complete, you should see a summary of all files which will be synchronized. Clicking the "Synchronize" button begins the backup process, which can take some time on the first run.

The next time you need to back up your data, you would connect your drive; unlock the encryption by entering the password; open FreeFileSync; and select the "Compare" button again. This time, you should only be presented the files which have been modified since the last backup. Then, the synchronization process should be much faster.

After I have conducted a backup, I open my email client and fetch all email from my provider onto my machine. I specifically do not do this before the backup. In the event that I deleted an important email from within my provider's server, which would then also delete the offline copy from my machine, I know that my previous backup has all emails which existed the previous week. This is minor, but something which has saved me in the past.

If I am able, I now shut down the machine and see how long I can go through the weekend without booting it back up. Some weekends are better than others. On Monday morning, I know I have a tidy, clean, and updated macOS machine ready for the week.

CONCLUSION

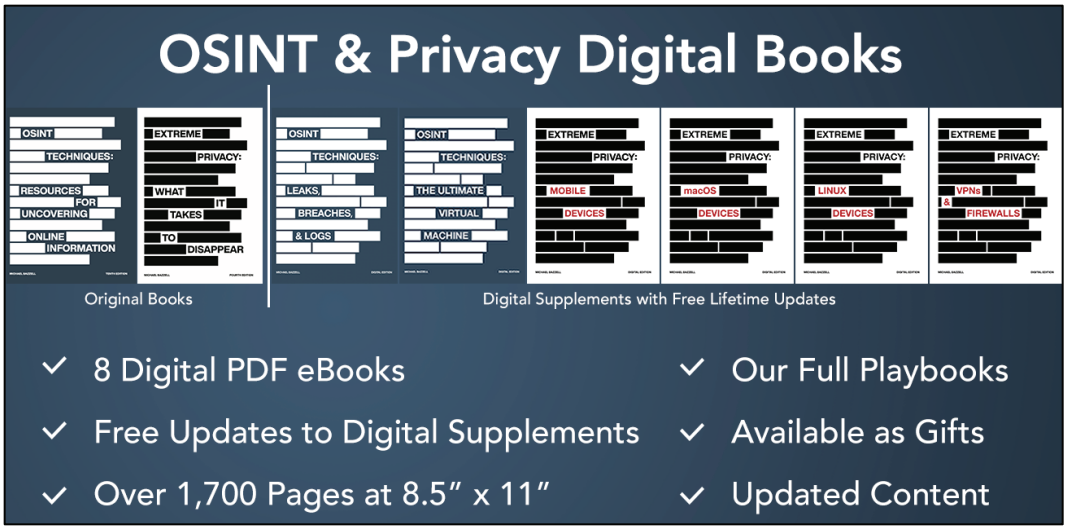
I hope you now possess a private and secure macOS device which does not share all of your activity with Apple. I believe you will find the minimalism and simplicity of your new device to be a superior experience. I practice what I preach, and configured my own macOS device from this guide. I no longer worry about unnecessary data collection or eavesdropping on my daily usage. My laptop no longer feels "dirty" a few weeks after using it. There is a great sense of freedom when you leave that world of data collection 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 $20 purchase supports the research which goes into creating and updating these guides.

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

~MB
IntelTechniques.com

Other Books by Michael Bazzell



OSINT Techniques, 10th Edition (2023): 36 chapters | 260,000 words | 550 pages | 8.5" x 11" | $30 - This textbook will serve as a reference guide for anyone who is responsible for the collection of online content. It is written in a hands-on style that encourages the reader to execute the tutorials while reading. The search techniques offered will inspire researchers to think outside the box when scouring the internet. Digital downloads include offline search tools, custom Linux scripts, and detailed report templates.

Extreme Privacy, 4th Edition (2022): 22 chapters | 320,000 words | 517 pages | 8.5" x 11" | $30 - This rewritten privacy manual is PROACTIVE. It is about starting over. It is the complete guide that I would give to any new client in an extreme situation. It leaves nothing out and provides explicit details of every step I take to make someone completely disappear, including legal documents and a chronological order of events. The information shared in this book is based on real experiences with my actual clients, and is unlike any content released in my other publications.

OSINT Techniques, Leaks, Breaches, & Logs (2024): 9 chapters | 57,000 words | 171 pages | 8.5" x 11" | $20 - This digital (PDF) supplement to ***OSINT Techniques, 10th Edition (2023)*** delivers a much more thorough guide about data Leaks, Breaches, & Logs. It provides our entire playbook which we use to locate, acquire, clean, store, and query various online data collections valuable to our investigations. All expired and outdated methods were replaced with new techniques, and brand-new topics were introduced throughout. We also explain all daily, weekly, and monthly tasks required to maintain your data collection. All updates are free and delivered digitally.

OSINT Techniques, The Ultimate Virtual Machine (2024): 14 chapters | 74,000 words | 231 pages | 8.5" x 11" | $20 - This digital (PDF) supplement to ***OSINT Techniques, 10th Edition (2023)*** delivers a much more thorough guide about Linux OSINT applications and Virtual Machines. It provides our entire playbook which we use to build, configure, clone, export, backup, and wipe out our investigative environments. All expired and outdated methods and applications were replaced with new functioning techniques, and brand-new programs were introduced throughout. For this release, we transition from Ubuntu to Debian

for a more private and stable environment, and also explain all tasks required to maintain your investigative machines. New downloadable scripts automate the entire installation process and all data search programs. All updates are free and delivered digitally.

Extreme Privacy, Mobile Devices (2024): 16 chapters | 73,000 words | 173 pages | 8.5" x 11" | $20 - This digital (PDF) supplement to ***Extreme Privacy, 4th Edition (2022)*** delivers a much more thorough guide about mobile devices. It provides our entire playbook which we use for our clients when we need to acquire new hardware, configure a custom operating system, execute proper DNS filtering, enable push services, install applications, obtain anonymous cellular service, establish VoIP connectivity, program redundant data eSIMs, provide secure communications, apply VPN strategies, and troubleshoot the things which will go wrong. We also explain all maintenance and best practices for a new private and secure device. All updates are free and delivered digitally.

Extreme Privacy, macOS Devices (2024): 10 chapters | 46,000 words | 126 pages | 8.5" x 11" | $20 - This digital (PDF) supplement to ***Extreme Privacy, 4th Edition (2022)*** delivers a much more thorough guide about macOS devices. It provides our entire playbook which we use for our clients when we need to sanitize previous Apple IDs; acquire new hardware; configure operating system settings; execute a proper firewall; install applications without Apple ID; configure browsers, VPNs, and DNS; establish VoIP connectivity; create virtual machines; and generate custom scripts for daily usage. We also explain all maintenance and best practices for a new private and secure macOS device. Purchase includes custom macOS scripts and an import file to replicate all firewall rules. All updates are free and delivered digitally.

Extreme Privacy, Linux Devices (2024): 10 chapters | 45,000 words | 119 pages | 8.5" x 11" | $20 - This digital (PDF) supplement to ***Extreme Privacy, 4th Edition (2022)*** delivers a much more thorough guide about Linux devices. It provides our entire playbook which we use for our clients when we need to acquire new hardware; configure operating system settings; execute proper DNS filtering; install applications securely; configure browsers and VPNs; establish VoIP connectivity; create virtual machines; and generate custom scripts for daily usage. We also explain all maintenance and best practices for a new private and secure Linux device. Purchase includes custom Linux scripts. All updates are free and delivered digitally.

Extreme Privacy, VPNs & Firewalls (2024): 9 chapters | 38,000 words | 101 pages | 8.5" x 11" | $20 - This digital (PDF) supplement to ***Extreme Privacy, 4th Edition (2022)*** delivers a much more thorough guide about VPNs and firewalls. It provides our entire playbook which we use for our clients when we need to acquire new hardware; configure firewall settings; execute proper DNS filtering; configure web browsers; and establish VPN connectivity. We also explain all maintenance and best practices for a new private and secure firewall device. Purchase includes custom firewall configuration files. All updates are free and delivered digitally.