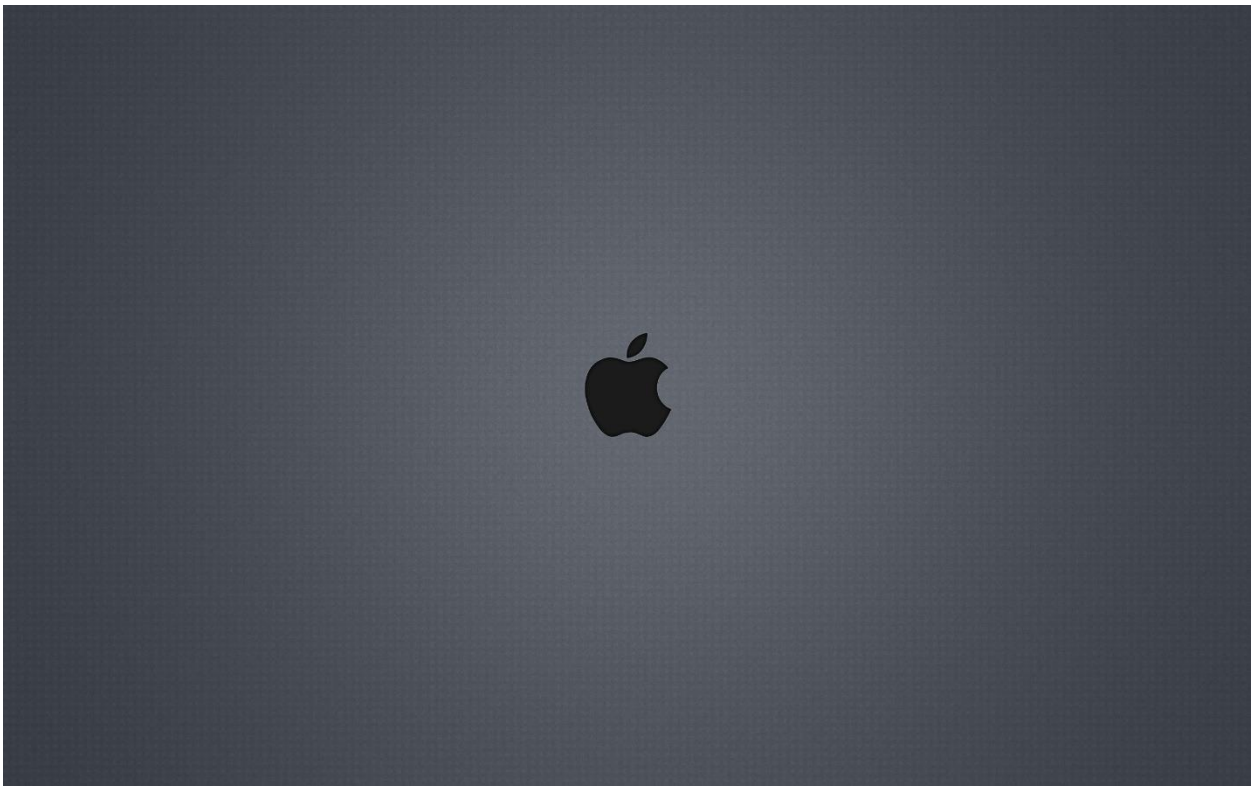
The Ultimate Guide to Hacking macOS



By **tokyoneon** 01/24/2019 4:34 am

Apple's macOS operating system is just as vulnerable to attacks as any Windows 10 computer or Android smartphone. Hackers can embed backdoors, evade antivirus with simple commands, and utilize USB flash drives to completely compromise a MacBook. In this always-updated guide, we'll outline dozens of macOS-specific attacks penetration testers should know about.

How Do I Hack a MacBook?

When thinking about this question, it's helpful to consider our proximity to the target Mac device. Our *distance* from the target device varies and gaining access increases in difficulty the further away we are. Whether we have physical access, share a Wi-Fi network with the device, or have enough information to remotely social engineer a user into opening a back doored application will determine the amount of effort an attacker needs to put forth to gain a remote shell.

With that in mind, the following articles are organized in an order based on the level of proximity from the target MacBook required to execute the hack.

We'll start by talking about the different kinds of attack vectors and payloads that can be crafted for macOS. Then, physical and USB flash drive attacks are discussed, followed by network-based attacks, and how macOS targets can be compromised from anywhere in the world. Finally, we'll look at exploits and major vulnerabilities the OS has suffered in recent years to give readers and bug bounty hunters an idea of where the operating systems most alarming issues have been discovered.

Hack macOS with a Single Command

Before we dive into executing payloads and social engineering targets, let's first have a look at a few commands that can be used to compromise the operating system.

Much like PowerShell in Windows 10, hackers abuse programming languages that come preinstalled in macOS. Many of these languages are powerful and allow for complex interactions with frameworks like EvilOSX, Empire, Bella, and Metasploit. In all of my tests, none of the featured one-liners were detected by macOS or antivirus software like Avast and AVG.

Using these built-in commands have been covered in the following articles.

1 Python Command to Bypass Antivirus

A Python payload is hosted on a remote server and ultimately executed on the target MacBook using a USB Rubber Ducky. It will take some suaveness to get the USB Rubber Ducky into the target Mac computer, but the rest is a cake walk.

- **Full Guide: Use One Python Command to Bypass Antivirus Software**

2 Ruby Command to Bypass Antivirus

For this hack, a Ruby payload is embedded into an AppleScript and crafted to look like an ordinary PDF file. The fake PDF is then shared with the intended target, who opens it up and gives us the win.

- **Full Guide: Use One Ruby Command to Bypass Antivirus Software**

3 Tclsh Command to Bypass Antivirus

The lesser-known, but very powerful Tcl command is used to evade antivirus software and backdoor a MacBook with just a few characters. The good thing about this attack is that it can handle abrupt backdoor disconnections.

- **Full Guide: Use One Tclsh Command to Bypass Antivirus Software**

Physically Hack a MacBook

By default, macOS is very vulnerable to physical compromises. Single-user mode attacks make it possible for an attacker to modify any file or directory as root — without a password or the target's knowledge. Even if the hard drive is encrypted with FileVault, it can be circumvented using password-guessing (brute-force) attacks. These methods have been covered in the below articles.

4 Recovery Mode to Extract & Brute-Force the Hash

Recovery mode is used to extract a login hash and later brute-forced with Hashcat to reveal the password in plain-text. The attacker could use a USB flash drive with another Mac computer to do the hard work or could simply create a temporary user on the target Mac instead.

- **Full Guide: How to Hack a Mac Password Without Changing It**

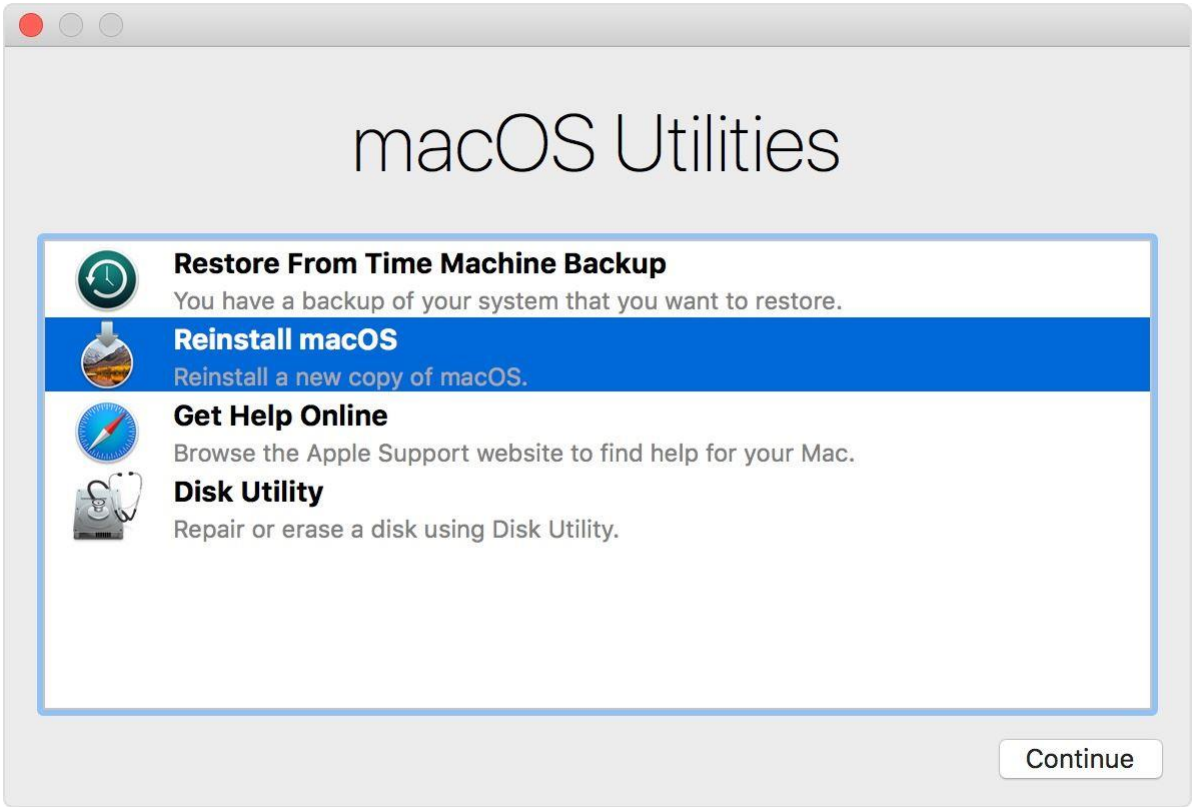


Image via Apple

5 Single-User Mode to Configure a Backdoor

Single-user mode is used to embed a Netcat listener into the target's device and executed using cron at set intervals. This method is most effective where the target device allows incoming connections and shares a Wi-Fi network with the attacker.

- **Full Guide: How to Configure a Backdoor on Anyone's MacBook**

6 Connect to Backdoors from Anywhere

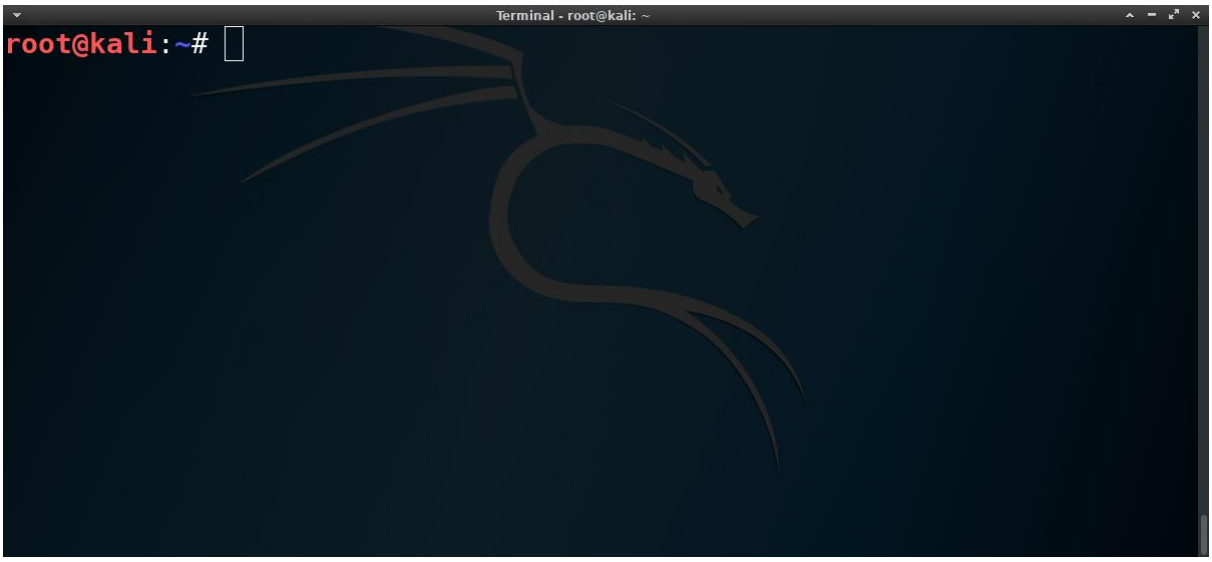
Improving upon the above article, this method completely evades the target's firewall and allows the attacker to control the MacBook as it moves between different Wi-Fi networks.

- **Full Guide: How to Connect to MacBook Backdoors from Anywhere**

7 Break Through FileVault Encryption

Some targets may have encrypted their hard drive with FileVault. While this will prevent the MacBook from being compromised in a few seconds, it's not entirely bulletproof. It's possible to automate a password-guessing attack against FileVault using the featured software and Bash script.

- **Full Guide: How to Break into a MacBook Encrypted with FileVault**



Get a Shell with Trojanized AppleScripts

Now, let's talk about embedding one-liner payloads into AppleScripts.

AppleScript, currently included in all versions of macOS, is a scripting language that allows users to directly control macOS applications, as well as parts of macOS itself. In each AppleScript application exists an embedded executable. This makes AppleScript applications easily one of macOS's most formidable attack vectors.

Normally, AppleScripts allow users can create harmless scripts to automate repetitive tasks, combine features from multiple legitimate applications, and create complex workflows. However, they can be abused by hackers to take control of a target's operating system.

8 Create a Fake PDF Trojan with AppleScript

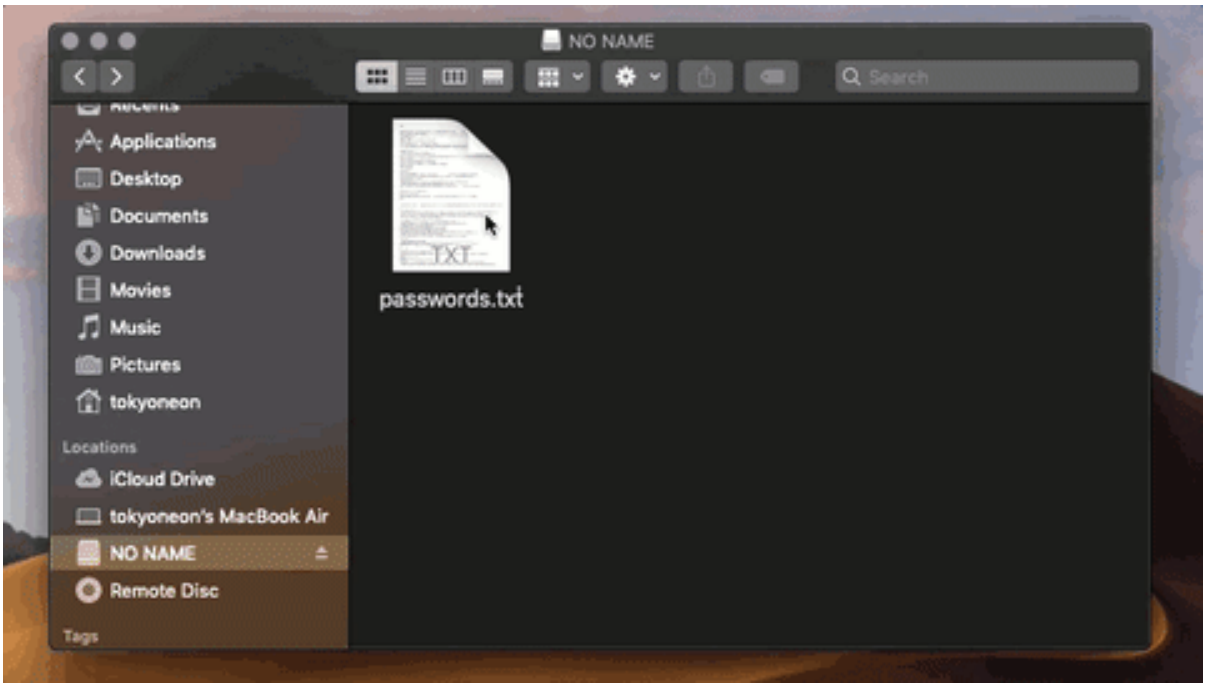
An introduction that covers creating an Empire stager intended for AppleScript trojans. The stager is later embedded into the AppleScript. After preparing a stager (or payload), the file extension and icon is spoofed to make the .app look like a genuine PDF.

- **Full Guide:** **How to Create a Fake PDF Trojan with AppleScript, Part 1**
- **Full Guide:** **How to Create a Fake PDF Trojan with AppleScript, Part 2**

9 Pretend to Be a Trusted App

AppleScript is used to circumvent the limitations of Mojave's new security features by social engineering the target into thinking a legitimate application is requesting administrative privileges.

- **Full Guide: Masquerade as a Trusted App to Bypass Elevated Privileges**



USB Drop Attacks

It may not always be possible to physically backdoor a MacBook. The next simplest method for compromising a target is social engineering them into opening trojanized AppleScripts. This can be accomplished by performing USB drop attacks, [which macOS is highly susceptible to](#).

Experiments have taken place that shows nearly 50% of people who find rogue USB flash drives insert them into their computer. This makes USB drop attacks an effective method for getting a shell without touching the target MacBook.

Shop USB Flash Drives on Amazon | Best Buy | Walmart

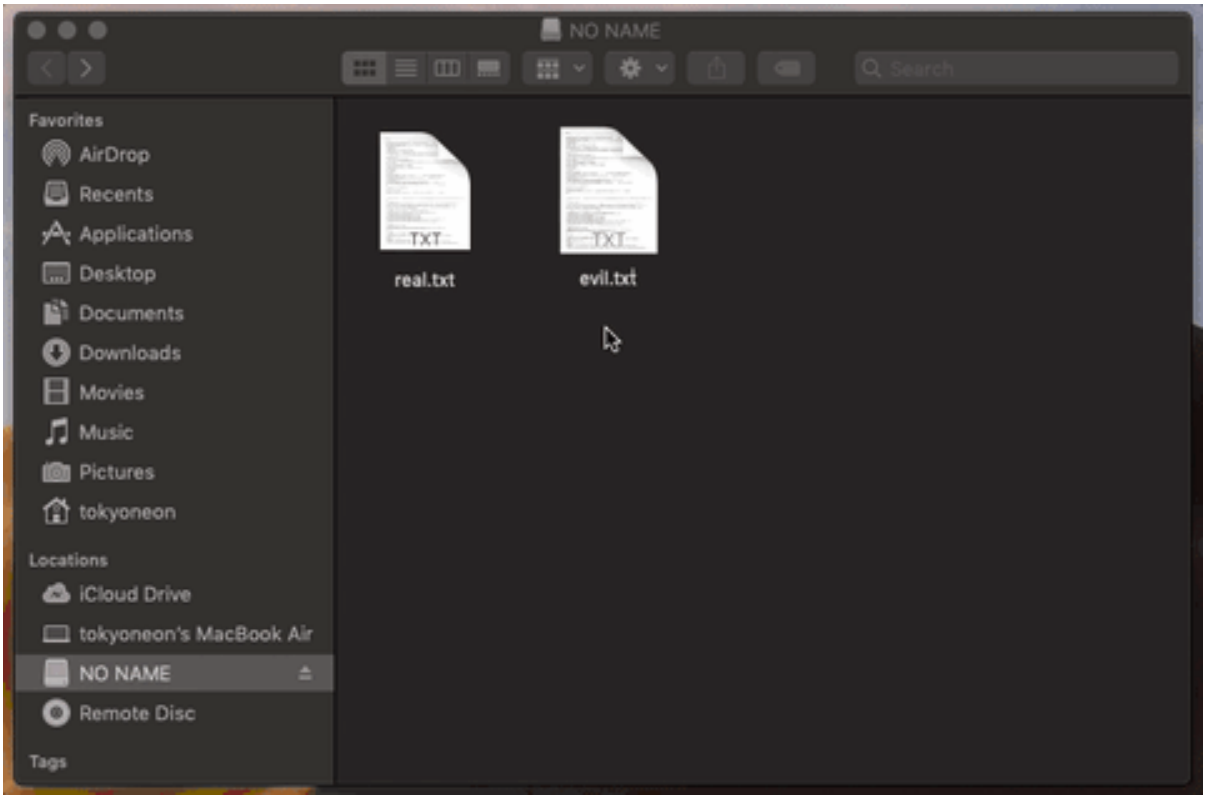
The USB flash drive containing AppleScript should be strategically placed somewhere the intended target will undoubtedly find it. This could be somewhere at their workplace, around their home, or by slipping it into their purse or backpack if close proximity possible.

USB drop attacks have been covered in the below articles.

10 Use a Self-Destructing Payload to Hack Mojave

Mojave's insecure USB file permissions allow for the execution of any kind of file or application — completely bypassing GateKeeper protections. This is exploited using an AppleScript disguised to look like a typical text file.

- **Full Guide: How to Hack Mojave with a Self-Destructing Payload**



11 Spread Trojans & Pivot to Other Mac Computers

Files found on a target's USB flash drive are modified and trojanized in an effort to remotely pivot from one Mac device to another.

- **Full Guide: How to Spread Trojans & Pivot to Other Mac Computers**

12 Use a Birthday Card to Hack Wi-Fi Passwords

While this article isn't geared toward macOS and focuses on capturing the target's Wi-Fi password, the social engineering aspect can be applied to a MacBook user. The use of a greeting card to trick a target into inserting an SD card or USB flash drive into their computer can be utilized in many different scenarios with many different goals.

- **Full Guide: How to Hack Anyone's Wi-Fi Password Using a Birthday Card**

Network-Based Attacks

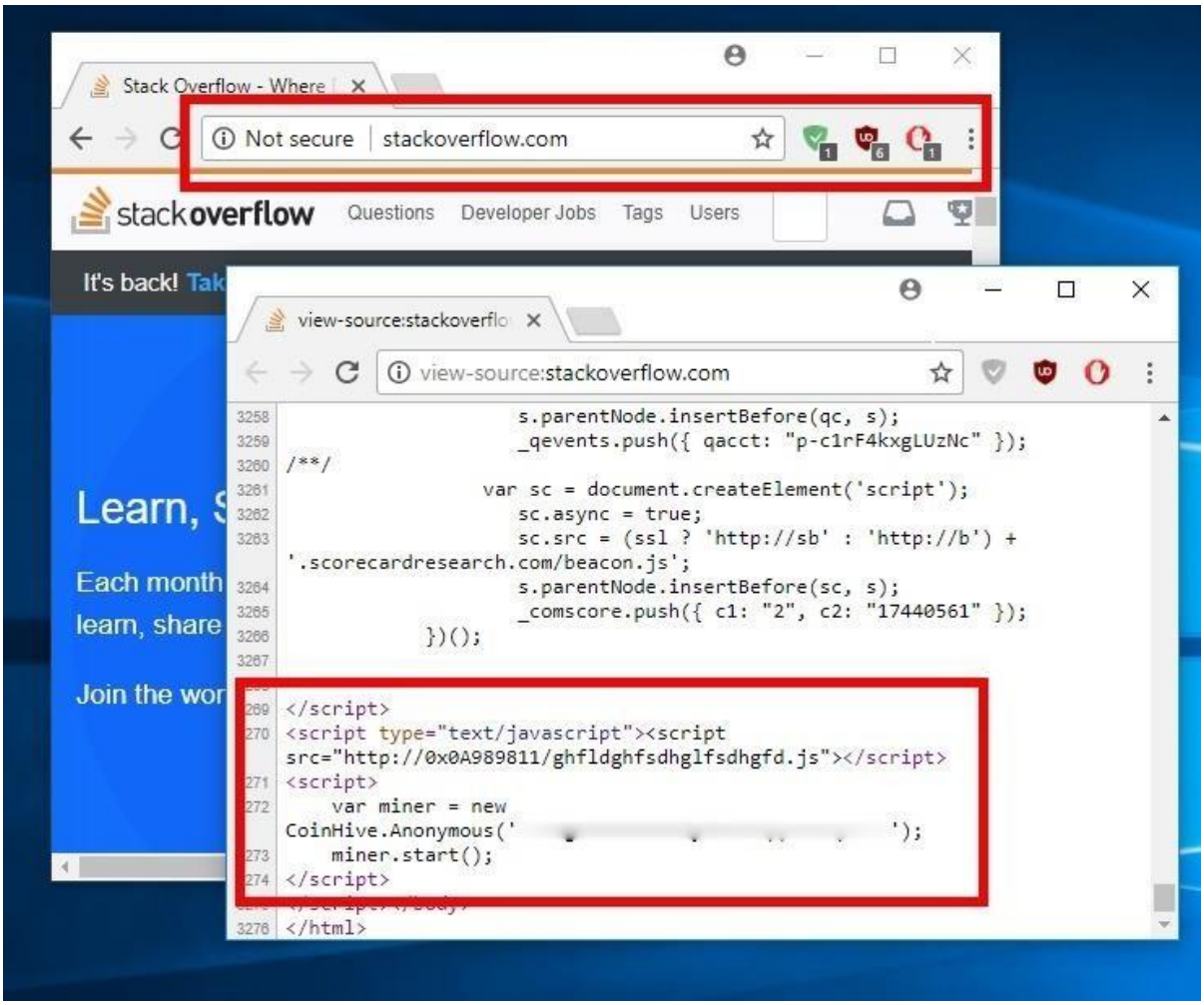
MacOS isn't immune to man-in-the-middle (MitM) or network-based attacks. Web traffic transmits between the MacBook and router just like any other internet-connected device. This traffic is easily manipulated and can be used to inject cryptocurrency miners into the target's web browser in real time.

Man-in-the-middle attacks have been covered in the following articles.

13 Inject Coinhive Miners into Public Wi-Fi Hotspots

A man-in-the-middle framework is used to inject JavaScript into the target's web browser. Additionally, this article covers URL obfuscation to evade some adblocker detections. This demonstrates how hackers compromise coffee shop Wi-Fi networks to force unsuspecting targets into mining cryptocurrency for them.

- **Full Guide: How to Inject Coinhive Miners into Public Wi-Fi Hotspots**



14 Flip Photos & Inject Messages into Browsers

Images in the target's web browser are manipulated in funny and obscene ways, through the use of a man-in-the-middle framework.

- **Full Guide: [Flip Photos, Change Images & Inject Messages into Friends' Browsers]**

15 Sniff Wi-Fi Activity without Connecting to the Router

Packets are captured and analyzed without connecting to the target Wi-Fi network. Like Windows 10 and smartphones, MacOS devices are affected and vulnerable to such attacks.

- **Full Guide: [Sniff Wi-Fi Activity Without Connecting to a Target Router]**

Hack a MacBook for Anywhere in the World

Hacking into MacBook's located in different parts of the world is a bit more involved than other methods mentioned in this article. GateKeeper, a security feature in macOS, is designed to keep shady AppleScripts from running in the operating system (shown below).

To protect users from malicious applications, a developer ID is required to sign applications and gain the "trust" of macOS in order for apps to execute. Unfortunately, for a fee, anyone with a credit card can acquire a developer ID and even share their malicious application using Apple's App Store.

App Store compromises first made headlines in 2015 when many apps were discovered exfiltrating user data to an attacker's server. And again, more recently, apps were removed from the App Store for stealing user data. And these are only the application removals that have been discovered or disclosed by independent security researchers. The true scale of this vulnerability is unknown. For all we know, Apple removes shady apps every day without informing the public.

It's certainly not impossible or very difficult to compromise macOS target's in different states or countries. It's a matter of being motivated enough to join Apple's developer ID program and simply paying for a certificate. In future posts, I may cover this topic in greater detail and update this section of the article.

Post-Exploitation Attacks

Commands and attacks executed *after* remote access has been established are classified as post-exploitation attacks. These attacks may consist of situational awareness, data exfiltration, covert desktop streaming, microphone eavesdropping, privilege escalation, and data dumping to name a few. I've covered many post-exploitation articles which are outlined below.

16 Perform Situational Awareness Attacks

A two-part article that demonstrates hardware and software enumeration, ARP cache dumping, locating sensitive files, and identifying connected storage mediums. After establishing a remote shell, it's important for an attacker to develop an understanding of their physical and network surroundings.

- **Full Guide: How to Perform Situational Awareness Attacks, Part 1**
- **Full Guide: How to Perform Situational Awareness Attacks, Part 2**

17 Install a Persistent Empire Backdoor on a MacBook

Netcat is used to elevate the functionality of the attacker's primitive backdoor to a fully-featured post-exploitation framework.

- **Full Guide: How to Install a Persistent Empire Backdoor on a MacBook**

18 Automate Screenshot Exfiltration

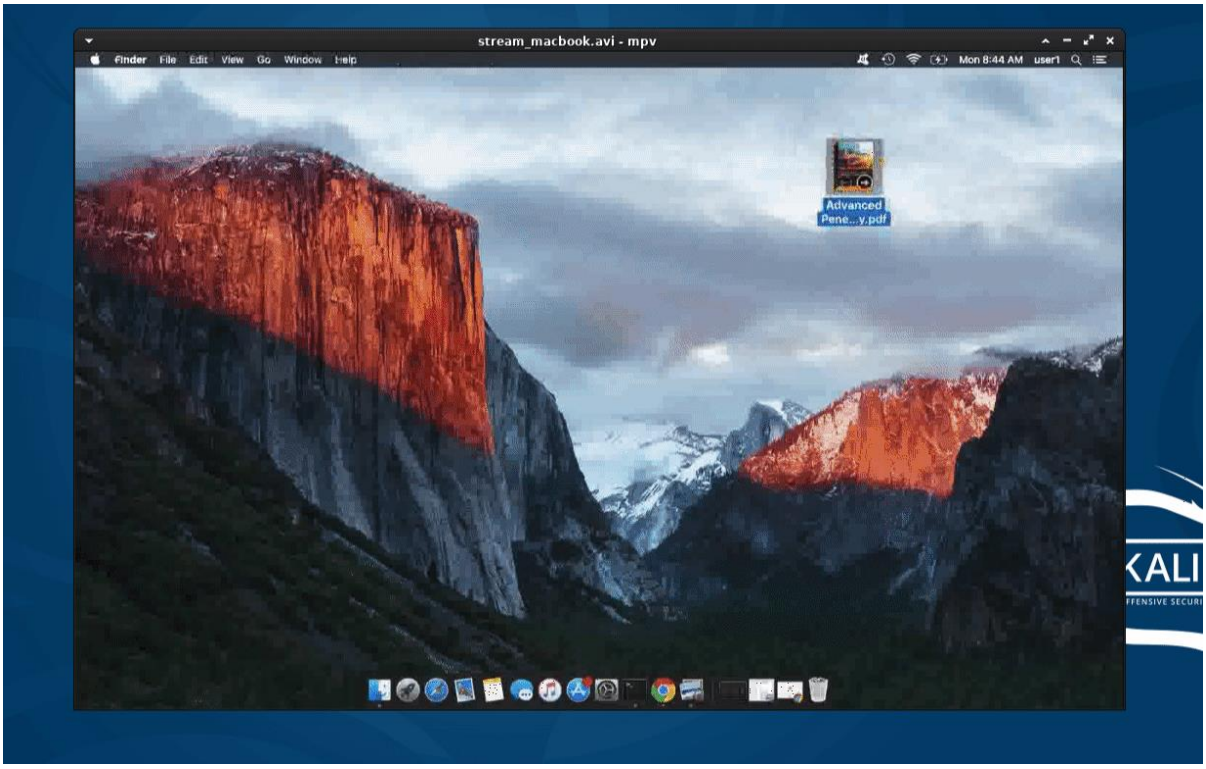
Screenshots of the target's desktop are taken covertly to passively observe behavioral activity. Such information can be used to further compromise the target and is commonly abused by blackhats for blackmail who have captured embarrassing or compromising conversations and photos.

- **Full Guide: Automate Screenshot Exfiltration from a Backdoored MacBook**

19 Secretly Livestream Someone's Screen Remotely

Taking the concept of observing behavioral activity to the next level, the entire MacBook screen is streamed to the attacker's computer and viewed in real time. This allows an attacker to see the target's every mouse click and keystroke without detection.

- **Full Guide:** [How to Secretly Livestream Someone's MacBook Screen Remotely]



20 Eavesdrop in Real-Time Using the Microphone

The MacBook's microphone is used to record conversations in the surrounding area and streamed to the attacker's system for real-time analysis.

- **Full Guide:** [Remotely Eavesdrop in Real Time Using a MacBook's Microphone]

21 Sniff Passwords on a Mac in Real Time

A two-part article showing how to stealthfully collect and exfiltrate a MacBook's unencrypted web traffic using a combination of tools such as Netcat, Empire, Tcpdump, Tshark, and Wireshark.

- **Full Guide:** [How to Sniff Passwords on a Mac in Real Time, Part 1]
- **Full Guide:** [How to Sniff Passwords on a Mac in Real Time, Part 2]

22 Dump Passwords Stored in Firefox Browsers Remotely

Passwords saved in Firefox are dumped using a low-privileged backdoor with just a few commands. With knowledge of the target's recent passwords, targeted wordlist attacks are possible and the macOS login password can be brute-forced.

- **Full Guide: How to Dump Passwords Stored in Firefox Browsers Remotely**

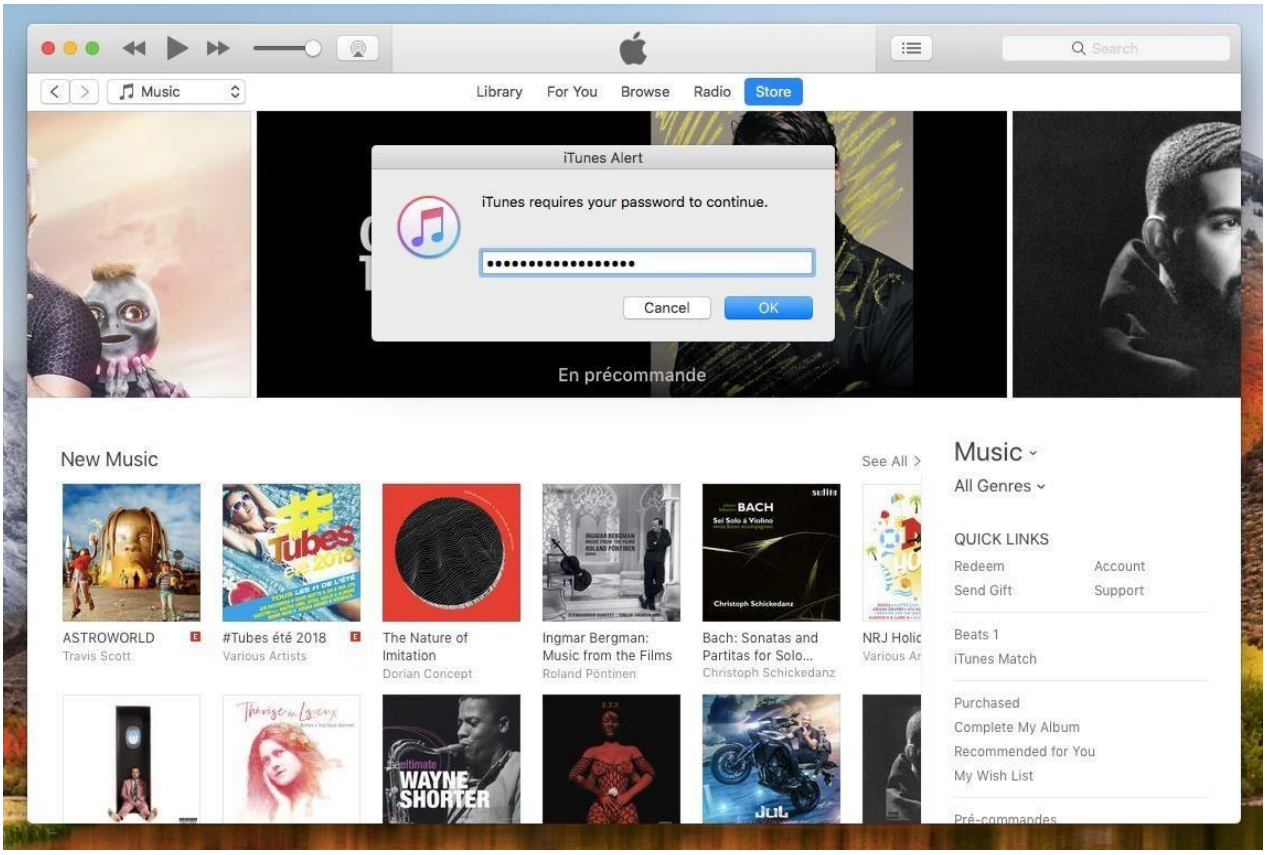
Privilege Escalation Attacks

It may be desirable to elevate the remote shell privileges to modify sensitive files and directories. Root privileges allow an attacker to execute commands with almost no security restrictions. There are several common methods of escalating to root privileges, as seen below.

23 Perform Privilege Escalation

Files owned by a root user are discovered to have overly permissive attributes and exploited by an attacker by embedding a backdoor. Alternatively, an Empire stager is used to invoke a credential phishing dialog to trick the target into divulging their login password.

- **Full Guide: How to Perform Privilege Escalation, Part 1 (File Permissions Abuse)**
- **Full Guide: How to Perform Privilege Escalation, Part 2 (Password Phishing)**



MacOS Zero-Days & Exploits

Hacking macOS doesn't end with the simple attacks outlined in this article. There are highly sophisticated macOS vulnerabilities and exploits currently deployed in the wild.

The term "zero-day" refers to code being used by hackers to quietly exploit unpatched vulnerabilities unknown to the software developer. MacOS has had its share of zero-days make news headlines in recent years.

For example, in 2017, Patrick Wardle disclosed a vulnerability that allowed unsigned applications to dump and exfiltrate a users Keychain with passwords in plain-text. In September 2018, Patrick disclosed a vulnerability that invoked virtual mouse-clicks without any user consent, which allowed an attacker to bypass any macOS security feature that relied on manual interaction with a notification dialog. Again, just a month later, Patrick revealed a vulnerability that completely bypasses Mojave's latest security features.

The three vulnerabilities just discussed were disclosed by just one individual. It's not unreasonable to believe a team of dedicated hackers is capable of finding similar exploits that have yet to be publicly disclosed.

Zero-days are in extremely high demand these days. Not by blackhats, but by cybersecurity companies and professionals. At the time of this writing, websites like Zerodium will pay up to $80,000 USD for a macOS or Safari exploit (as shown below). Other bug bounty programs offer millions of dollars for a single zero-day.

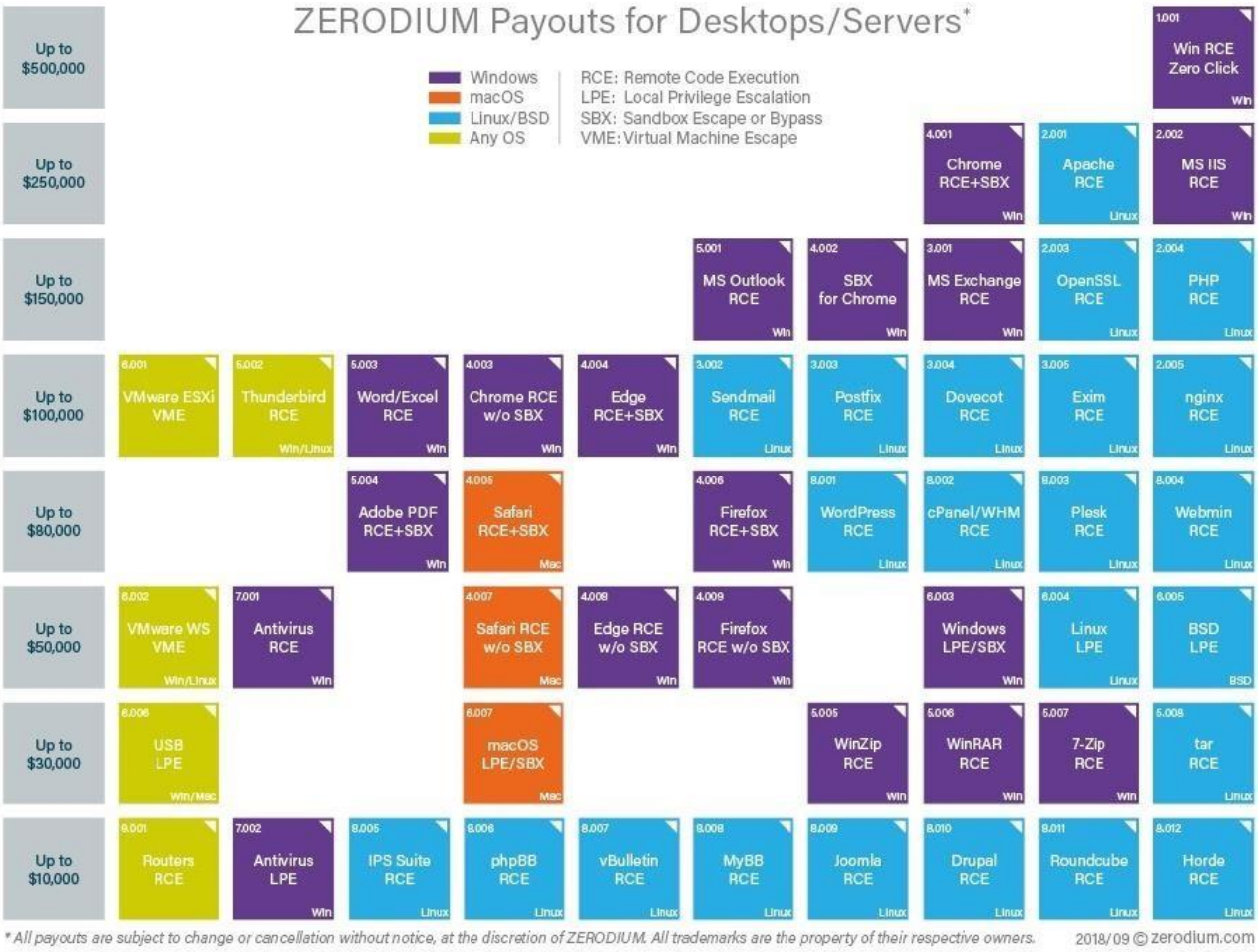


Image via [Zerodium]

And it doesn't end there. Vulnerabilities and application-specific exploits that don't make news headlines have turned up on [Exploit Database] at least once a month for the last few years. There were nearly [40 disclosed vulnerabilities in 2017] alone including [local privilege escalation], memory disclosure, and [arbitrary file execution] exploits.

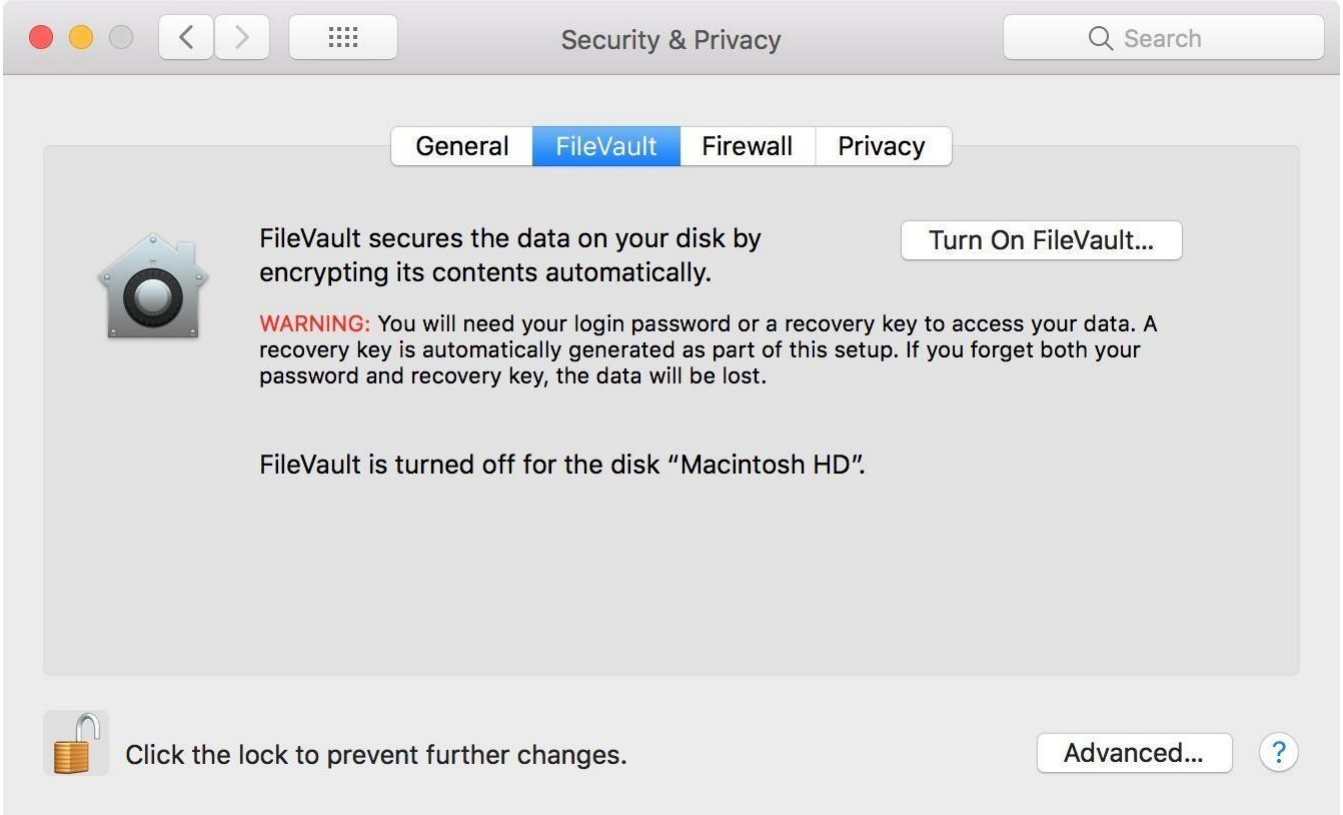
How to protect yourself from macOS Attacks

In truth, there are many ways of compromising a macOS device. Below are a few things readers can do to identify and prevent such activity from happening to them.

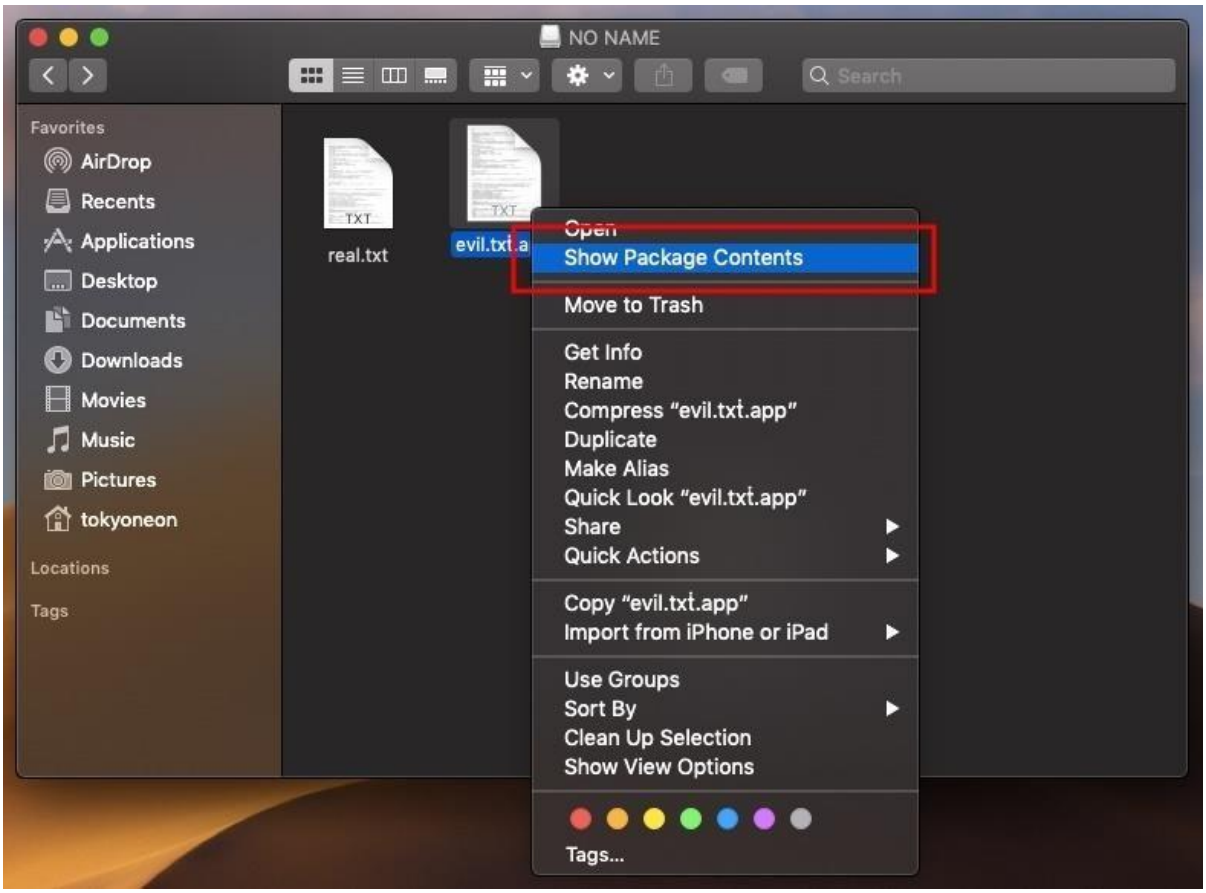
- **Don't use strange USB flash drives**. If you've stumbled upon a USB flash drive that doesn't belong to you, don't use it. This is possibly the best advice we can give. A well-placed USB flash drive could be the result of a well-thought-out social engineering attack against you or your employer. Regardless of how the USB flash drive is labeled or what files appear to be on it — don't insert it into your MacBook.

- **Enable firmware password protection**. To prevent attacker's from booting into a live USB, single-user mode, or recovery mode, set a firmware password. The firmware will only prompt for an additional password at boot if someone attempts to boot the MacBook into single-user, startup manager, target disk, or recovery modes. A firmware password, however, will not protect the hard drive in the event the disk is physically removed from the MacBook.

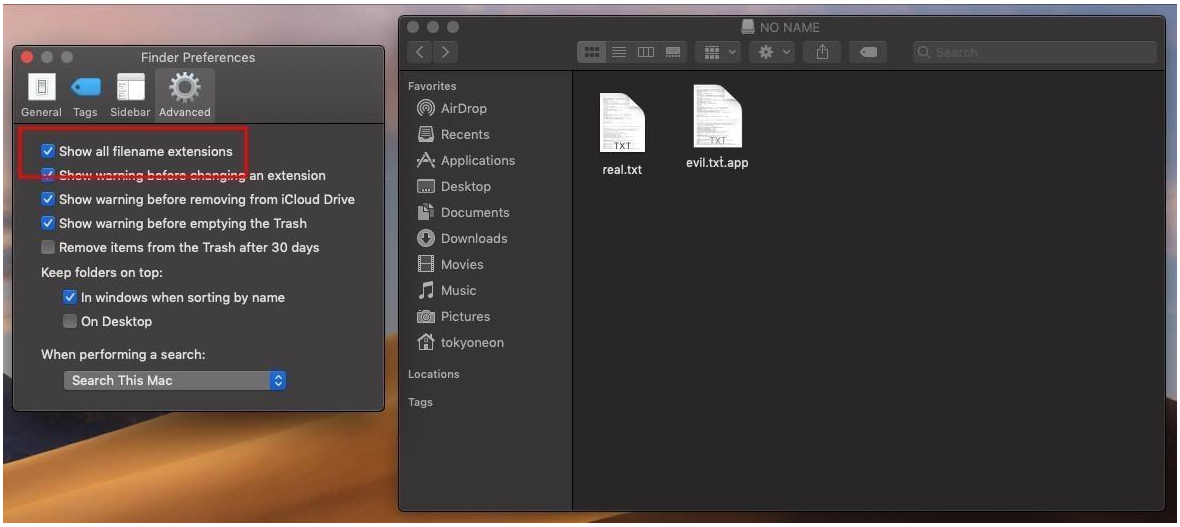
- **Enable FileVault encryption**. FileVault can be enabled by navigating to "System Preferences," then "Security & Privacy," and clicking "Turn On FileVault" (you may need to unlock the settings first). When it completes, the MacBook will restart and require a password to unlock the computer every time the Mac starts up. No account will be permitted to login automatically and accessing single-user mode will also require a password. This is the best way to prevent attacks against the encrypted disk even if it's physically removed from the laptop. A complex passphrase over 21 characters in length is recommended to protect against attackers with dedicated brute-force hardware.



- **Don't double-click files**. It's always best to explicitly choose which program to use when opening files. Right-click on the desired file, and manually choose an application from the "Open With" menu. You'll notice AppleScript applications don't have an "Open With" option in the context menu. This is because they're actually directories and can't be opened with applications such as TextEdit.



- **Show all filename extensions**. The Unicode trick used to spoof file extensions only works if "Show all filename extensions" is disabled, which it is by default. Enable this setting by navigating to "Finder" in the menu bar, then "Preferences," and check the option under the "Advanced" tab. The .app file extension will be forced and cannot be spoofed.



- **List files on the USB flash drive**. When in doubt, use the Terminal to list (**ls**) files on the USB flash drive. File extensions can't be spoofed here under any circumstances. Use this command with **-l** to print the USB flash drive's contents in a list format, and **-a** to show all files on the USB flash drive, including hidden files.

  ```
  ls -la /Volumes/USB-NAME-HERE/

  drwxrwxrwx@ 1 tokyoneon  staff    16384 Sep 28 11:01 .
  drwxr-xr-x@ 4 root       wheel      128 Sep 28 03:52 ..
  drwxrwxrwx@ 1 tokyoneon  staff    16384 Sep 28 05:03 evil.txt.app
  -rwxrwxrwx  1 tokyoneon  staff  3566129 Sep 28 02:40 real.txt
  -rwxrwxrwx  1 tokyoneon  staff  1938446 Sep 28 02:41 .hiddenfile.txt
  ```

- **Check for suspicious files**. Startup daemons and directories used by macOS include /Library/LaunchDaemons, /Library/LaunchAgents, and /Users/<username>/Library/LaunchAgents. Files in these directories can be inspected by opening Terminal, using the **cd** and **ls** commands to change into the desired directory and view its contents. The launchctl command can be used to disable any suspicious daemons and removed using the rm command.

- **Use private browser mode**. Dumpzilla can do a lot more than just extract passwords from Firefox. It's safer to use private browser mode 100% of the time. Though it may be inconvenient and make browsing the internet painful, it's actually quite dangerous to entrust so much data to web browsers. Browser data dumps containing dozens of email addresses and passwords are shared freely in black hat hacking communities. If hackers aren't selling your data, they're wreaking havoc on your accounts for amusement because it has no financial worth to them.

- **Use a master password**. If saving passwords in Firefox is a convenience you're not willing to give up, use a strong master password. This will provide a moderate obstacle for hackers and may prevent them from learning all of your passwords.

- **Use a proper password manager**. Password managers offer improved protection of stored passwords. Hackers can still exfiltrate and perform brute-force attacks against the password manager's database but with a strong and unique password, attacker's will have to spend weeks (or months) trying to crack the encrypted database.

That's it for now. We'll continue to update this roundup when we discover new macOS attack vectors. And until next time, follow me on Twitter @tokyoneon_. And be sure to leave a comment below if you have questions!