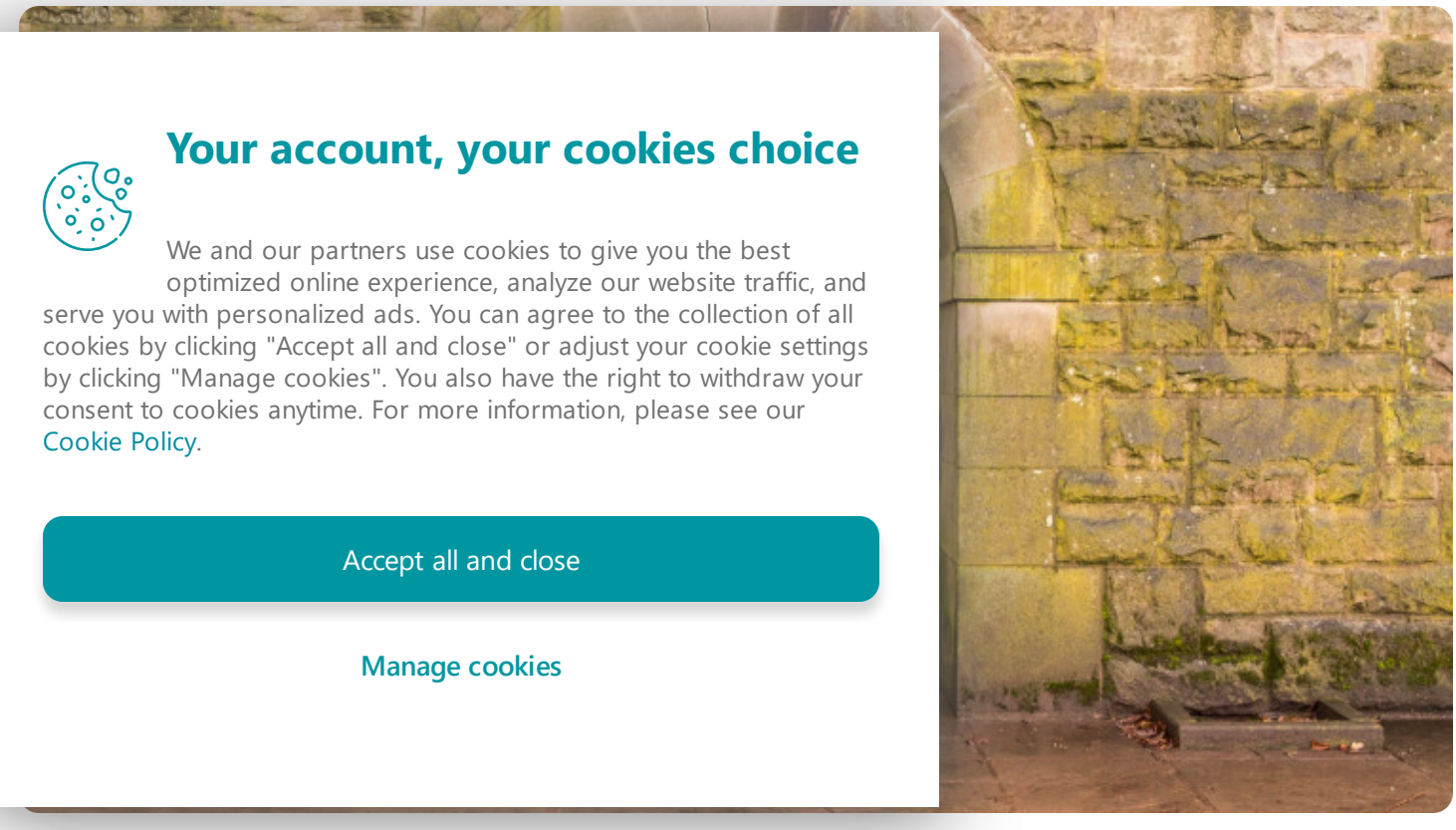
ESET RESEARCH

OceanLotus: macOS malware update

Latest ESET research describes the inner workings of a recently found addition to OceanLotus's toolset for targeting Mac users

Romain Dumont

09 Apr 2019 , 7 min. read



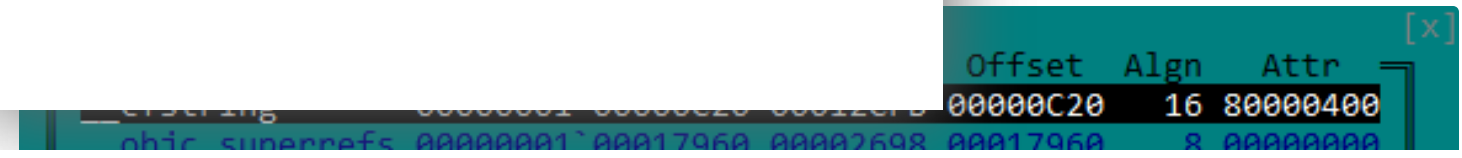
Early in March 2019, a new macOS malware sample from the OceanLotus group was uploaded to VirusTotal, a popular online multi-scanner service. This backdoor executable bears the same features as the previous macOS variant we looked at, but its structure has changed and its detection was made harder. Unfortunately, we couldn't find the dropper associated with this sample so we do not know the initial compromise vector.

We recently published a detailed update about OceanLotus and how its operators employ a wide range of techniques to gain code execution, achieve persistence, and leave as little trace as possible on a Windows system. OceanLotus is also known to have a malicious macOS component. This article details what has changed from the previous macOS version analyzed by Trend Micro and describes how, while analyzing this variant's code, you can automate string decryption using the IDA Hex-Rays API.

Analysis

The following three sections of this blogpost describe the analysis of the sample with the SHA-1 hash `E615632C9998E4D3E5ACD8851864ED09B02C77D2`. The file is named flashlightd and is detected by ESET products as OSX/OceanLotus.D.

ed with UPX, but most packer
ause they mostly include a
urther, Mach-O signatures are
haracteristic makes static
ng is that the entry point is
`TEXT` segment. This section has



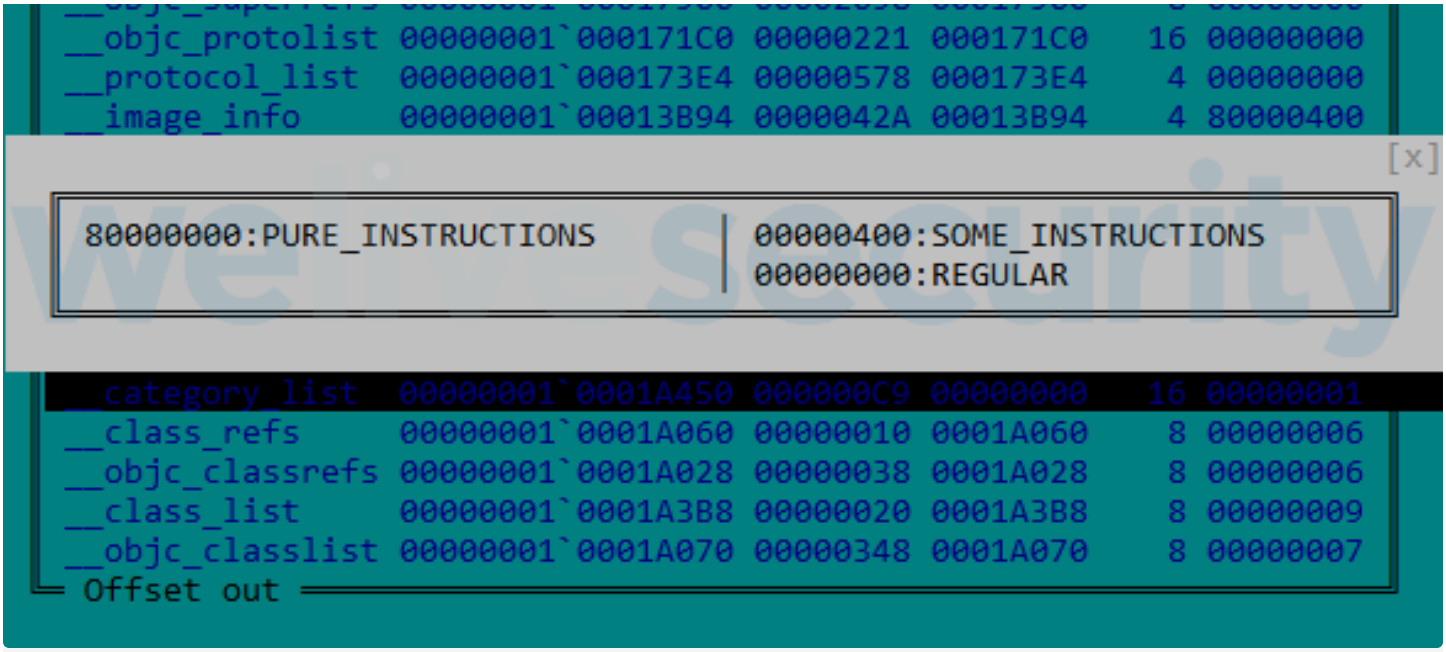
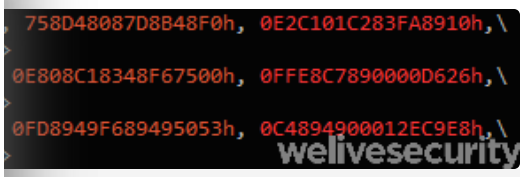


Figure 1. MACH-O `__cfstring` section attributes

As seen in Figure 2, the fact that the code is in the `__cfstring` section tricks some disassembly tools to display the code as strings.



ata by IDA

gging watchdog whose sole
order to do that, this thread:

`_ATTACH` as a request parameter

- Checks if some exception ports are open by calling the `task_get_exception_ports` function

- Checks if a debugger is attached, as seen in Figure 3, by verifying if the `P_TRACED` flag is set in the current process

```
LABEL_6:
    info.kp_proc.p_flag = 0;
    mib[0] = CTL_KERN;
    mib[1] = KERN_PROC;
    mib[2] = KERN_PROC_PID;
    mib[3] = getpid();
    size = 0x288LL;
    sysctl(mib, 4u, &info, &size, 0LL, 0LL);
    v1 = (unsigned __int16)(info.kp_proc.p_flag & P_TRACED) >> 11;
```


Figure 3. Check if a debugger is attached via `sysctl` function

If the watchdog detects that a debugger is present the `exit` function is called. Moreover, the sample then checks its environment by issuing the following two commands:

`ioreg -l | grep -e "Manufacturer"` and `sysctl hw.model`

vn virtualization system strings:

d:

`| awk '/Boot ROM Version/`

"MB", "MM", "IM", "MP" and "XS".

e, "MBP" stands for MacBook

nce the Trend Micro article, we

r this sample are quite recent as

- daff.faybilodeau[.]com
- sarc.onteagleroad[.]com
- au.charlineopkesston[.]com

The URL resource used has changed to `/dp/B074WC4NHW/ref=gbps_img_m-9_62c3_750e6b35`.

The first packet that is sent to the C&C server contains more information regarding the host machine. All data gathered by the commands in the following table are included.

Commands	Description
system_profiler SPHardwareDataType 2>/dev/null - awk '/Processor / {split($0,line,":"); printf("%s",line[2]);}'	Gather processor information
- awk '/Memory/ {split($0,line,":"); printf("%s",line[2]);}'	Gather memory information
- ifconfig en0 | awk '/ether/ {print $2}' - ifconfig en1 | awk '/ether/ {print $2}'	Gather network interface MAC addresses
- awk '/IOPlatformSerialNumber/ { split($0, line, "\"");	Retrieves the serial number of the device

```
printf("%s", line[4]); }'
```

On top of this configuration change, this sample does not use the libcurl library for network exfiltration. Instead, it uses an external library. To locate it, the backdoor tries to decrypt each file in the current directory using AES-256-CBC with the key `gFjMXBgyXWULmVVVzyxy` padded with zeroes. Each file is "decrypted" and saved as `/tmp/store` and an attempt to load it as a library made using the dlopen function. When a decryption attempt results in a successful call to dlopen, the backdoor then retrieves the exported functions `Boriry` and `ChadylonV`, which seem to be responsible for the network communication with the server. As we do not have the dropper or other files from the original sample's location, we could not analyse this library. Moreover, since the component is encrypted, a YARA rule based on these strings would not match the file found on disk.

As described in the analysis of the group's previous macOS backdoor, a *clientID* is created. This identifier is the MD5 hash of the return value of one of the following commands:

latformSerialNumber/ {

latformUUID/ { split($0,

he MAC address)

uidgen" in the previous samples

the return value indicating root

le System/HFS/25cf5d02-

code runs as root, or in

s/drivers/snippets.ecgML

ction and its timestamp is

modified using the `touch -t` command with a random value.

String decryption

Like previous variants, the strings are encrypted using AES-256-CBC (hex-encoded key: `9D7274AD7BCEF0DED29BDBB428C251DF8B350B92` padded with zeroes and the IV is filled with zeroes) using the `CCCrypt` function. The key has changed from previous versions but since the group is still using the same algorithm to encrypt strings, decryption could be automated. Along with this article, we are releasing an IDA script leveraging the Hex-Rays API to decrypt the strings present in the binary. This script may help future analysis of OceanLotus and the analysis of existing samples that we have not yet been able to obtain. At the core of this script lies a generic method to obtain the arguments passed to a function. Moreover, it looks for the parameter assignments in order to find their values. This method could be reused to retrieve the list of arguments of a function and then pass them to a callback.

Knowing the prototype of the *decrypt* function, the script first finds all cross-references to this function, finds all the arguments, decrypts the data and puts the plaintext inside a comment at the address of the cross-reference. In order for the script to work correctly, the [illegible] be set in the script and the global [illegible] s a DWORD in this case; see


```
74h, 0ADh, 7Bh, 0CEh, 0F0h, 0DEh, 0D2h, 9Bh
                ; DATA XREF: f_MachineCheck+49↑o
                ; f_MachineCheck+424↑o ...
 28h, 0C2h, 51h, 0DFh, 8Bh, 35h, 0Bh, 92h

                ; DATA XREF: f_MachineCheck+3E↑o
y               ; f_CheckMachineType+4B↑o ...
```

```
y_len
```

In the *Function* window, you can right-click the decryption function and click “Extract and

decrypt arguments". The script should put the decrypted strings in comments, much as in Figure 5.

```
00000001000012C3 movaps  xmm0, xmmword ptr cs:a_vmware ; "Y?:zC??????^?"
00000001000012CA movaps  [rbp+var_50], xmm0
00000001000012CE mov     [rbp+var_40], 0
00000001000012D2 lea     r12, key_len
00000001000012D9 mov     ecx, [r12]
00000001000012DD lea     rbx, key
00000001000012E4 lea     rdi, [rbp+var_50]
00000001000012E8 mov     esi, 10h
00000001000012ED xor     r8d, r8d
00000001000012F0 mov     rdx, rbx
00000001000012F3 call    f_decrypt       ; vmware
00000001000012F8 mov     r13, rax
00000001000012FB movaps  xmm0, xmmword ptr cs:a_virtualbox ; "????????\x15???\x7F????"
0000000100001302 movaps  [rbp+var_70], xmm0
0000000100001306 mov     [rbp+var_60], 0
000000010000130A mov     ecx, [r12]
000000010000130E lea     rdi, [rbp+var_70]
0000000100001312 mov     esi, 10h
0000000100001317 xor     r8d, r8d
000000010000131A mov     rdx, rbx
000000010000131D call    f_decrypt       ; virtualbox
0000000100001322 mov     r15, rax
0000000100001325 movaps  xmm0, xmmword ptr cs:a_oracle ; "-?????,?l\x05A?????"
000000010000132C movaps  [rbp+var_90], xmm0
0000000100001333 mov     [rbp+var_80], 0
0000000100001337 mov     ecx, [r12]
000000010000133B lea     rdi, [rbp+var_90]
```

```
... p  f_MachineCheck+B9     call   f_decrypt; oracle
... p  f_MachineCheck+F0     call   f_decrypt; parallels
... p  f_MachineCheck+155    call   f_decrypt; ioreg -l | grep -e "Manufacturer"
... p  f_MachineCheck+43A    call   f_decrypt; sysctl hw.model
... p  f_CheckMachineType+67 call   f_decrypt; system_profiler SPHardwareDataType 2>/dev/null | awk '/Boot ROM Version/
... p  f_CheckMachineType+16C call  f_decrypt; MBP
... p  f_CheckMachineType+1D4 call  f_decrypt; MBA
... p  f_CheckMachineType+23C call  f_decrypt; MB
... p  f_CheckMachineType+2A4 call  f_decrypt; MM
... p  f_CheckMachineType+30C call  f_decrypt; IM
... p  f_CheckMachineType+374 call  f_decrypt; MP
... p  f_CheckMachineType+3D8 call  f_decrypt; XS
```

Figure 6. Xrefs to of `f_decrypt` function

The final script can be found on our Github repository.

Conclusion

As recently documented in another of our articles, the OceanLotus group keeps improving tools for targeting Mac users. ac users don't run security of less importance. ESET the network library used for the t network protocol used remains

ibutes, are also available in our

Domain names

- daff.faybilodeau[.]com
- sarc.onteagleroad[.]com
- au.charlineopkesston[.]com

URL resource

- /dp/B074WC4NHW/ref=gbps_img_m-9_62c3_750e6b35

File paths

- ~/Library/SmartCardsServices/Technology/PlugIns/drivers/snippets.ecgML
- /Library/Storage/File System/HFS/25cf5d02-e50b-4288-870a-

	ESET detection name
C77D2	OSX/OceanLotus.D

Tactic	ID	Name	Description
	T1158	Hidden Files and Directories	The backdoor hides the *clientID* file via **`chflags`** function.
	T1107	File Deletion	The backdoor can receive a "delete" command.
Defense Evasion	**T1222**	File Permissions Modification	The backdoor changes the permission of the file it wants to execute to 755.
	T1027	Obfuscated Files or Information	The library used for network exfiltration is encrypted with AES-256 in CBC mode.
	T1099 (macOS)	Timestomp	The timestamp of the file storing the clientID is modified with a random value.

ckdoor performs a fingerprint of the
e on its first connection to the C&C

ckdoor encrypts the data before
tion.

ckdoor implements a specific format
packet involving random values. See
Micro article.

Related Articles

CH

earch Podcast: CosmicBeetle

ESET RESEARCH

Embargo ransomware: Rock'n'Rust

Discussion