How to Perform Situational Awareness Attacks, Part 2

By **tokyoneon** 09/15/2018 12:46 am

It's important to know whom you're dealing with after hacking your target's MacBook. Getting remote access is simple, but covertly gathering information about the user and their system can be a challenge.

Continuing with our situational awareness attacks, we'll be further orienting ourselves within a compromised Mac device and deepening our knowledge of the target's behavioral activities. We can accomplish this by using tools built right into macOS (previously called Mac OS X).

With these tools, we can pillage the target's Terminal history for previously run commands, find recently modified files containing sensitive information, and identify external hard drives and USB drives for pivoting attacks. Armed with this information, we can develop a profile of a target's activities and further exploit them and their network.

- **Previously: Situational Awareness Attacks, Part 1 (Using System Profiler & ARP)**

1 Find Interesting Files & Directories

When issuing commands from a remote backdoor, the convenience of Spotlight and Finder are not accessible to us. But there other ways to get the information we want.

Find is an incredibly powerful tool that can be abused by hackers. It allows us to locate specific files with great ease. There are far too many arguments to demo in one article, so I'll instead show a few examples readers can build on.

The below **find** command will recursively search the Downloads/ directory for a files (**-type f**) with the PDF (**"*.pdf"**) extensions. The wildcard (*) instructs find to locate *any* kind of PDF, regardless of its file name.

```
find /Users/<username>/Downloads/ -type f -name "*.pdf"
```

Next, a slightly more advanced find command is used, seen below. This time, find will search the Documents/ directory for files using multiple (**\(... \)**) file extensions ending

with ".sh" and ".txt" — discovered text files and Bash scripts will be displayed in the terminal.

```
find /Users/<username>/Documents/ -type f \( -name "*.sh" -o -name "*.txt" \)
```

It might be helpful to an attacker to know which files the target user has modified in the last *X* minutes. The below find command will search every file and directory in the tokyoneon/ home folder for files modified (**-mmin**) in the last 5 minutes.

```
find /Users/tokyoneon/ -mmin -5
```

Many of the files discovered in the Application Support/ and Preferences/ directories are not compelling, as seen in the below output. But recently modified files in Documents/ or on the Desktop/ may prove useful to an attacker looking for ways of pivoting or escalating their privileges.

```
/Users/tokyoneon//Desktop
/Users/tokyoneon//Desktop/.DS_Store
/Users/tokyoneon//Desktop/important_credentials_2018.rtf
/Users/tokyoneon//Library/Application Support
/Users/tokyoneon//Library/Application Support/com.apple.spotlight.Shortcuts
/Users/tokyoneon//Library/Application Support/com.apple.sharedfilelist
/Users/tokyoneon//Library/Application
Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentDocuments.sfl2
/Users/tokyoneon//Library/Application
Support/com.apple.sharedfilelist/com.apple.LSSharedFileList.RecentApplications.sfl2
/Users/tokyoneon//Library/Application Support/AddressBook/Metadata/.info
/Users/tokyoneon//Library/Preferences
/Users/tokyoneon//Library/Preferences/ByHost
/Users/tokyoneon//Library/Preferences/ByHost/com.apple.loginwindow.75294026-DDFE-5804-9373-
CA2196D2BD6E.plist
/Users/tokyoneon//Library/Preferences/com.apple.spaces.plist
/Users/tokyoneon//Library/Preferences/com.apple.commerce.plist
/Users/tokyoneon//Library/Preferences/com.apple.AddressBook.plist
/Users/tokyoneon//Library/Preferences/com.apple.textInput.keyboardServices.textReplacement.plist
/Users/tokyoneon//Library/Preferences/com.apple.xpc.activity2.plist
/Users/tokyoneon//Library/Preferences/com.apple.systemuiserver.plist
/Users/tokyoneon//Library/Preferences/com.apple.Spotlight.plist
/Users/tokyoneon//Library/Containers/com.apple.TextEdit/Data/Library/Saved Application State
/Users/tokyoneon//Library/Containers/com.apple.TextEdit/Data/Library/Preferences
/Users/tokyoneon//Library/Containers/com.apple.TextEdit/Data/Library/Preferences/com.apple.TextEdit.pl
ist
/Users/tokyoneon//Library/KeyboardServices/TextReplacements.db
/Users/tokyoneon//Library/KeyboardServices/TextReplacements.db-wal
/Users/tokyoneon//Library/KeyboardServices/TextReplacements.db-shm
/Users/tokyoneon//Library/Caches/GeoServices/networkDefaults.plist
```

We can further refine this kind of file discovery using the **-mtime** argument. This will allow us to identify files older than *X* days old — but also files not older than *Y* days. The **+** and **-** characters are used to say *older than* and *not older than*, respectively. So, a command using **+1** and **-60** will instruct find to locate files older than 1 day but also not older than 60 days.

```
find /etc/ -mtime +1 -mtime -60
```

In my example above, I'm searching the etc/ directory. These files can't be modified by low privileged users without a password, but we can still read their contents which may contain sensitive configuration information.

```
/etc/auto_master
/etc/krb5.keytab
/etc/aliases.db
/etc/asl
/etc/asl/com.apple.authd
/etc/rtadvd.conf
/etc/auto_home
/etc/ppp
/etc/php-fpm.d
/etc/localtime
/etc/newsyslog.d
/etc/pam.d
/etc/pam.d/chkpasswd
/etc/pam.d/login.term
/etc/pam.d/screensaver
/etc/pam.d/checkpw
/etc/pam.d/authorization
/etc/pam.d/login
/etc/pam.d/passwd
/etc/apache2
/etc/apache2/httpd.conf
/etc/apache2/original
/etc/autofs.conf
/etc/ssh
/etc/resolv.conf
/etc/ntp.conf
```

2 Examine the Target's Terminal History

Some tech-savvy macOS users might find themselves using the Terminal for a variety of things. For example SSH and FTP commands are often executed using Terminal. These commands may reveal additional servers the target macOS user has access to. Terminal

commands can also help us build behavioral profiles that can be used for future social engineering attacks.

If you're a macOS user, you can open a Terminal and use the **history** command to view recently executed commands. This can also be viewed from a Netcat backdoor.

Below is a Terminal output example after running **history**.

```
history

    1   nc 192.168.1.44 55555
    2   sudo su
    3   sudo su
    4   sudo su
    5   cd Desktop/
    6   cat 3.scpt
    7   ls -la /System/Library/Launch*
    8   ssh -X -p 4441 -i /Users/tokyoneon/office/ssh/main root@4.3.2.1
    9   sudo su
   10   ifconfig en0
   11   cd /tmp/unix-privesc-check-master/
   12   ./upc.sh --help
   13   ping 192.168.1.21
```

On line 1, we can see some Netcat activity which might be worth further investigating. But more importantly, on line 8, there's SSH login information. The **-i** argument used in the SSH command indicates there's an SSH private key called "main," located in the /office/ssh/ directory. This can potentially be used by an attacker to compromise a remote server operated by the macOS user. In business and corporate environments, such exposures can be catastrophic.

Alternatively, if there are multiple users on the system, we may be able to use the **cat** command to read a specific user's Terminal history. These history files are usually located in each users home directory with the ".bash_history" file name.

```
cat /Users/<username>/.bash_history
```

For a more comprehensive and tactical approach, we can locate every single Bash history file on the device and **cat** each file automatically.

```
find / -type f -iname *bash_history* -exec cat {} \;
```

For the above command, find will search every single directory (**/**) for a file (**-type f**) containing "bash_history" anywhere in the name (**-iname**). It will then execute (**-exec ... {} \;**) the **cat** command to print their contents.

3 Identify External Disks & USBs

USB flash drives attached to the macOS device are excellent vectors for spreading malicious files. The concept is simple: the target inserts their own USB thumb drive into the compromised MacBook, and the attacker remotely swaps files on the USB device for payloads that appear to be ordinary files.

If the USB flash drive is then shared between colleagues and other computers, the attacker would effectively pivot to other devices. This attack is explained in greater detail in my "Spread Trojans & Pivot to Other Mac Computers" article.

There are several ways of enumerating attached USB device information. We can quickly **ls** the /Volumes/ directory, but for a more thorough approach, we'll use the **diskutil** and **system_profiler** commands.

There will likely be several disks displayed with the below command.

```
diskutil list
```

We're looking for disks labeled "external, physical" in the output below. This indicates an external hard drive, SD card, or USB drive is attached to the macOS device. Based on its size (15.5 GB), the device is most likely a 16 GB USB flash drive.

```
/dev/disk3 (external, physical):
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     FDisk_partition_scheme                        *15.5 GB    disk3
   2:             Windows_FAT_32 KINGSTON                15.5 GB    disk3s1
```

We can also use the **system_profiler** command with the **SPUSBDataType** argument here for a much more granular look at attached devices.

```
system_profiler SPUSBDataType
```

In the below output, we now know the target is using a USB 3.0 DataTraveler by Kingston which is FAT32-formatted. This means the USB flash drive is cross-platform and can be used between Windows, macOS, and Linux operating systems. More importantly, the USB flash drive is mounted to the OS with write permissions, which means we can modify and add malicious files to the device as needed.

```
USB:

    USB 3.0 Bus:

      Host Controller Driver: AppleUSBXHCIPPT
      PCI Device ID: 0x1e31
      PCI Revision ID: 0x0004
      PCI Vendor ID: 0x8086

        DataTraveler 3.0:

          Product ID: 0x1666
          Vendor ID: 0x0951  (Kingston Technology Company)
          Version: 1.00
          Serial Number: XXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Speed: Up to 5 Gb/sec
          Manufacturer: Kingston
          Location ID: 0x14400000 / 4
          Current Available (mA): 900
          Current Required (mA): 504
          Extra Operating Current (mA): 0
          Media:
            DataTraveler 3.0:
              Capacity: 15.5 GB (15,502,147,584 bytes)
              Removable Media: Yes
              BSD Name: disk3
              Logical Unit: 0
              Partition Map Type: MBR (Master Boot Record)
              USB Interface: 0
              Volumes:
                KINGSTON:
                  Capacity: 15.5 GB (15,498,018,816 bytes)
                  Available: 15.36 GB (15,359,361,024 bytes)
                  Writable: Yes
                  File System: MS-DOS FAT32
                  BSD Name: disk3s1
                  Mount Point: /Volumes/KINGSTON
                  Content: Windows_FAT_32
                  Volume UUID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

Digging Deeper...

MacOS has a plethora of built-in tools that hackers can abuse to acquire information about the target and their system. We can still discover and pivot to additional user accounts, identify and evade installed antivirus software, and fingerprint services running

on the device for exploitation. The information gathering possibilities are vast and we've only scratched the surface.