The Worm in the Apple

An Analysis of Modern OSX Malware

Author:
WALTON LEE
Walton.lee@tufts.edu

Mentor:
MING CHOW

December 15, 2015

Abstract

For years, OSX malware have largely employed the same techniques for initial downloading and persistence, with the most successful malicious applications mainly utilizing social engineering to infect computers. As such, detecting and preventing attacks have become trivial tasks for OSX developers having come up with a suite of built-in defenses for basic attack vectors. However, the recent boost in Mac sales has made OSX a more valuable target and developers have begun to design creative exploits to get past OSX's built-in security. This paper addresses the recent developments in OSX malware and the updated procedure to defend against the revitalized threat, which requires the right third-party software and the user's ability to identify threats by understanding how malware penetrates computer systems.

1 Introduction

The typical Mac user gives little consideration to the issue of computer security, believing viruses are problems that are exclusive to Windows and Unix users. Even Apple themselves have helped spread this perception through their popular "I'm a Mac" ad campaign. However, the lack of viruses in the past decade has been largely attributed to Mac computers cornering only a small percentage of the computing market share, thus being a less attractive target for black-hat crackers. In recent years, mac computers have become more popular than ever, dominating the laptop market and enjoying increased popularity due to branding and design factors. As a result, more malware targeted towards OSX have appeared on the market. Bit9 and Carbon Black have described 2015 as "The most prolific year in history for OSX malware" in a report, stating that OSX malware is five times more prevalent this year than the preceding five years featuring notable viruses such as Flashback, iWorm and MacDefender.[i]

2. To the Community

Since Mac computers are now ubiquitous, it has become increasingly important for the average mac user to gain a working knowledge of the kinds of malware that can potentially threaten his/her computer. Some more technology savvy members of the Mac community may believe that simply browsing the web while utilizing safe browsing practices, avoiding suspicious third-party downloads and handling passwords cautiously, is enough to prevent malware attacks when using a Mac machine. However, this perception is false; there are many layers to social engineering and preventing all malicious downloads onto a machine is virtually impossible, barring staying disconnected. Moreover, malware attacks in the recent past have become more advanced and have involved less of social engineering aspect, meaning malware that doesn't require a human mistake may very well be available soon. For those who want to stay ahead of the technological curve and be prepared for when that time comes, please continue reading.

3 Conventional Malware

Before any malware can run and wreak havoc on a computer, it must first be downloaded and then installed. Malware installation on OSX computers can be accomplished through a variety of techniques, some of which are Mac specific but most can be applied to any machine. Drive-by downloads are the most common method of getting malicious software initially downloaded onto a computer. By visiting a malicious site, a rogue html or JavaScript script will initiate a malware download. Other types of malware known as Trojans use social engineering to deceive users into installation. One kind of Trojan software disguises itself as a free download for beneficial or expensive software. The MacDefender program poses as anti-malware software but in reality steals credit card information[ii]. Another type of Trojan pretends to be a patch/update/plug-in for frequently used software. Flashback is a Trojan that infected hundreds of thousand computers by appearing to be a flash or java plugin for web browsers.

Once malware has been downloaded, it must be installed before it can unleash its payload. There is a common misconception that malware installation requires the user to enter an admin password. If malware has been previously installed on the machine, there are a number of default applications that can be exploited to automatically run malware. The most exploitable of which is the Launchd service for OSX. Launchd manages processes known as LaunchDaemons and LaunchAgents to automatically run applications when a mac either boots up or logs in a user. If malware that takes advantage of these processes has been downloaded, it can use Launchd to continually run it itself, possibly installing even more malware without the need of admin authorization. Almost every malware in circulation make use of these persistence exploits to increase their impact and longevity. Login items and startup items are similar processes that are equally exploitable but due to their ability to be directly accessed from system preferences, are less commonly used.[iii]

Even if installed malware doesn't affect an OSX machine directly, the malware can still use the machine to infect other platforms. Many windows viruses can proliferate via Mac computers. By sending infected emails or sharing infected files with windows-users, Mac users can cause Windows users to accidentally download harmful software since the download is coming from a supposedly trusted source. Malware can also spread by way of hardware. Many types of malware are capable of transferring using USB or even the Thunderbolt media port e.g. Thunderstrike. In the past couple years, a breed of malware capable of infecting iOS devices via connection to infected OSX computers have cropped up. Dubbed "Wirelurker", this malware when connected to an iOS device will install malicious applications on the device, which are capable of stealing personal information such as texts and contact information.

4 Built-in Mac Defenses

All OSX machines come bundled with a number of standard defenses. These built-in security protocols are designed to stop malware

at different steps of infection with the intended result of having a comprehensive multilayered security suite.

The first layer of defense affects Internet connection. OSX has four programs designed to prevent malicious downloads from the web: Gatekeeper, XProtect, the application firewall and App Transport Security. Gatekeeper is a program that checks all third-party software downloads for a digital signature signifying approval by the Apple Store. Gatekeeper can also be set to only allow downloads from the Apple store. It may also be turned completely off. XProtect is a very rudimentary anti-virus application, screening all download names for a list of blacklisted malware names. Due to its basic behavior, the amount of security XProtect provides is very minimal. Next is the application firewall, which blocks incoming connections for unauthorized applications. The application firewall is turned off by default however and will require users to manually activate. Finally there is App Transport Security, which imposes a series of security practices on all apps, most notably requiring applications to connect to the Internet via secure protocols (HTTPS).

The next layer of security involves applications and code running locally on the OSX machine. This layer includes perhaps the most important security feature provided by OSX, which is Sandboxing. Sandboxing is the process by which OSX limits the rights of applications to read and access other files. Ideally with this defense, malicious web threats or infected applications are rendered unable to steal and manipulate sensitive data. The most recent update of OSX, El Capitan, Apple has also introduced System Integrity Protection, a security policy that applies to all code; only the Apple Installer and System Update processes can run system binaries, protecting essential utilities used for system administration.

The final layer of protection is the hardware layer. OSX provides the ability to encrypt the entire startup disk when the computer is shut down, preventing any forced extraction of data. It must be set up manually but fortunately, hardware attacks are the least likely attack vector if users are sensible i.e. they don't plug in random memory storage devices to their computer. [iv]

5 Advanced Mac Malware

Due to the offensive nature of malware, there are existing techniques that can be used to bypass the OSX security suite with a variety of tools targeting each of OSX's security processes. The restrictive nature of the OSX firewall and application transport security typically limits online downloads via web browsers to be the most likely source of initial infection. Unfortunately, obvious malware downloads such as video codecs and browser plug-ins may not be the only malicious downloads users have to lookout for. Any file downloaded over http is vulnerable to a man-in-the-middle attack, where an attacker can intercept the connection between a client and server then modify any data travelling between the two. More sophisticated malware can take advantage of these attacks and insert themselves into any download that isn't protected by SSL, which is the vast majority of

online downloads including many trusted antivirus software (AV).

The two most relevant mac defenses to preventing initial malware downloads are Gatekeeper and Xprotect. Xprotect, being an extremely simple antivirus program, can easily be bypassed simply by renaming any components of the virus that are on Xprotect's blacklist. Gatekeeper on the other hand, is a more sophisticated program and getting around it involves a more involved exploit called dylib hijacking. Dylib hijacking the OSX version of a widely used window exploit called dll hijacking where attackers would plant libraries with the same name and structure as existing system libraries but would contain malware. These libraries would be loaded into applications that require system libraries of the same name but reach the false directories first when looking for the appropriate files. Hijacking dylibs, the dynamic library extension on OSX, similarly exploits the logic by which applications loads libraries. There is a multitude of ways to do this but one straightforward example is by exploiting a logic error in OSX load commands where applications can fail to load "weak" or unnecessary libraries but will load them if they are available. Some malware can simply plant its payload in a directory labeled as a missing library and the payload will be executed whenever the application is run. Bundled with an apple approve app, the download will have no problem getting pass Gatekeeper. Dylib hijacking also provides an alternate persistence method to the usual login items, which are easy to detect and remove.[v]

Once installed, most kinds of malware will attempt to access personal data about the user from other running applications or files. This is exactly what OSX's sandboxing mechanism aims to prevent. Unfortunately, as with the aforementioned defenses, there are a variety of methods to bypass this safeguard. The first security threat involves Keychain, an in-built OSX app that manages all the users credentials by storing keys, passwords and certificates associated with each app, with each app having an access-control list (ACL) that determines which other apps and directories it can interact with. If malware is already running, it can simply pose as a pre-existing version of a target application and once the target stores a password, it will recognize the malware's keychain as its own and share give its password away. All applications along with their helper-program each also have a bundle ID (BID) and the Apple store makes sure no two apps share the same BID. However, it doesn't do the same for helper programs so malicious applications can contain helper-programs that share BIDs with those of real applications. When a targeted application is downloaded, the operating system will recognize that a preexisting BID exists and adds the target application to the ACL of the malicious app, giving the malware complete access to the newly downloaded app. This exploit is called container cracking. The next attack is IPC interception. Many browsers and applications use protocols such as Websocket to establish a single-socket connection with other local applications on a predetermined port. Malicious applications, using the standard permissions they are given, can hijack these ports and listen in on any data passing through. The last example is called scheme hijacking. All applications contain a scheme URL, which helps determine which applications data gets passed to. For example

whenever an **http** scheme is called, the machine's default Internet browser will open. When two applications share the same scheme, the app downloaded earlier on OSX and the app downloaded later on iOS will be where the data gets passed. The exploit is obvious – give a malicious application a URL scheme that is identical to that of an application the user has not downloaded. These attacks are all collectively referred to as cross application resource access attacks (XARA) and expose the failings of OSX's most critical security feature.[vi]

Since the aforementioned attacks largely exploit subtle vulnerabilities the operating logic, many of these attacks will be able to bypass El Capitan's new System Integrity Protection. Moreover most commercial anti-malware software can be easily evaded by either writing new malware that make use of the same exploits or even just recompiling existing versions such that they produce a different signature that is not recognized by behavioral based anti-malware. A final interesting exploit is that even firewalls cannot block outgoing connections to iCloud. Thus, even firewalls both native and commercial are incapable of blocking file exfiltration e.g. the sending of files containing sensitive data to remote iCloud account by malicious applications.

6 Defenses

As mentioned above, most anti-malware software, even paid ones, can easily be bypassed due to their use of traditional scanning techniques such as signature checking. Moreover, the techniques outlined above provide methods to not only get around every preexisting OSX defense but also the means to fool smart users who browse the Internet with security-aware practices. So how does one protect their Mac? The answer involves using software that is conscious of how modern Mac malware works and knows where to scan for malicious files as a result.

Patrick Wardle, director of R&D and Synack, presents Objective-see – a collection of anti-malware software that detects malware in places other commercial anti-malware services don't know to look – which includes KnockKnock, Blockblock, Task-explorer and Dylib Hijack Scanner. The cutely named Knock-knock and BlockBlock work together to target persistent software. Knockknock is a scanner that checks OSX computers for persistently installed software, flagging any known malware and filtering Apple-signed binaries. Blockblock, on the other hand, alerts the user know if any persistent software has been newly installed. Task-explorer displays all running processes, allowing the user to filter by digital signature, view all libraries used by a process and monitor any network connections created by a task. This allows users to scan their running tasks for suspicious software that either call on strange libraries or access unnecessary files. Dylib Hijack Scanner rounds out the software suite, scanning for applications that are either susceptible to dylib hijacking or have already been hijacked. All of the applications in the collection work in tandem with the Virustotal online service to scan files and application for known mal-ware.[vii]

At first glance, the Objective-see software suite appears to be the perfect solution to the growing OSX malware problem. However, the use of these applications comes with one subtle prerequisite: working knowledge of modern OSX malware. Even though the suite employs virustotal, it is only capable of detecting well-known malware. It is up to the user to manually filter through the lists of persistent software and dynamic libraries to find undocumented malware. The risks of using these apps without a good understanding of how malware attacks work include unintentionally deleting non-infected applications, shutting-down useful processes or removing crucial system libraries. Hopefully this paper will have provided the necessary insight to begin using these programs to protect your computer.

7 Conclusion

At the start of this paper we established the fundamental ideas employed by all OSX malware and the standard safeguards OSX uses to provide minimum security. We then showed how modern malware could employ more elegant techniques to subvert OSX's natural defenses and ended with a range of technologies that when managed properly can prevent even the cleverest attacks. While it is possible that Mac users will be fine simply by relying on OSX's built-in security and utilizing safe web-browsing practices, it is hard ignore the growing threat of OSX malware and the variety of techniques readily available to malware developers. Mac users also bear the responsibility of preventing the spread of malware to Windows and OSX platforms alike; simply remaining bug-free on one's own computer is not enough for security-minded individuals. In order to maintain a completely secure computer, users need to understand the various aspects of malware, namely initial attack vectors, persistence techniques and cross-application referencing. With this knowledge, users can correctly identify threats to their computer that would otherwise slip past automated defenses, third-party or native.

References

[i] Bit9 & Carbon Black. 2015: The Most Prolific Year in History for OSX Malware. https://assets.documentcloud.org/documents/2

[ii] Chester Winiewski. Mac Users Hit with Fake AV When Using Google Image Search https://nakedsecurity.sophos.com/2011/05/02/mac-users-hit-with-fake-av-when-using-google-image-search/, December 2015.

[iii] Patrick Wardle. Writing Bad-@$$ Malware https://www.blackhat.com/docs/us-15/materials/us-15-Wardle-Writing-Bad-A-Malware-For-OS-X.pdf, December 2015.

[iv] Safety. Built right in. http://www.apple.com/osx/what-is/security/, December 2015

[v] Patrick Wardle. Dylib Hijacking on OSX https://www.virusbtn.com/virusbulletin/archive/2015/03/vb201503-dylib-hijacking, December 2015

[vi] Luyi Xing, Xiaolong Bai, Tongxin Li, Xiaofeng Wang, Kai Chen, Xiaojing Liao. Unauthorized Resource Access on Mac OSX and iOS http://www.netfast.com/wp-content/uploads/2015/06/Apple-Zero-Day-Threat-Research.pdf, December 2015

[vii] Patrick Wardle. Objective-see https://objective-see.com/index.html, December 2015.