The Evolution to Fileless Malware

David Patten

East Carolina University

Abstract

Malware and viruses have been around since the early days of computers. The computer security industry has often played a game of cat and mouse with malware authors in which the malware authors create new and complex malware programs and the security industry develops better programs to protect and prevent malware. Recently malware has taken a new approach to attacking computers, fileless malware that does not rely on writing complex malware programs. This new fileless malware depends on commonly installed programs to cause damage and extract information. This paper provides a look at the evolution malware followed by an examination of the use of .NET Framework, PowerShell, offensive PowerShell tools, and PowerShell Forensic Tools.

Keywords: Malware, .Net Framework, PowerShell, Fileless Malware, Virus, Information Security.

The Evolution to Fileless Malware

Malware, a thorn in the side of modern society, has been around since the earliest days of computers; however, malware has evolved. The early days of malware required a talented malware author spending hours writing assemble code to serve their purpose. Since then, malware has evolved into a cybercrime industry that is full of cybercriminals looking to sell malware and use malware to generate a profit. The task of creating malware has become easier as higher level programming languages have been adopted. The .Net framework has enable developers to create new categories of malware that through the use of PowerShell create malware that is fileless in nature. This paper will examine the history of malware by dividing it into 5 categories: Early Malware, Windows Malware, Network Worms, Ransomware, and State

Sponsored Malware. Following the history of malware this paper will look at the evolution of malware to the fileless malware that is beginning to become prevalent. This examination will include explanation of the .NET framework, PowerShell, various offensive PowerShell tools, and PowerShell Forensic Tools. The paper will end with a conclusion looking at the current state of malware and computer security.

History of Malware

The history of malware can be divided into several categories that are defined by the time period and technology used in the category or significant changes in technology. The first category is the beginning of malware. During the early days of malware, it was transmitted primarily through floppy disks. One of the early viruses, Brain.A, was created by two brothers, Basit and Amjad, from Pakistan. The malware was created as a proof of concept to show the insecurities in the early days of personal computers. The virus simply installed itself on the boot sector of floppy disks that were inserted into an infected computer and infected computers when inserted. The malware did no harm but had the contact information of the virus authors and proved a point regarding early computer security. (Milošević, 2013) Much of the early malware was proof of concepts that were created in the new computer industry that was just beginning to emerge.

Phase two is the early Windows phase. During this phase, the first malware to affect Microsoft Windows was released. The first Windows virus, WinVir, was similar to the Brain.A virus in that it did little harm. WinVir's main feature was its ability to replicate. The malware would create copies of itself and then remove itself from the original infected file. It was a self-replicating and self-destructing virus. Additionally the first mail worm and macro worm were created as well. The first macro virus was WM.Concept. This piece of malware was written for

Microsoft Word in 1995. The malware was spread through sharing documents. The WM.Concept worked on both Macintosh and PC computers that had Microsoft Word installed. The malware used the macro function inside Microsoft Word to copy a malicious template over the master template and thus infecting each new document created on the computer. (Milošević, 2013) Another notable piece of malware from this era is the Happy99 virus. Happy99 was the first email virus and while it was a fairly benign the virus that when executed would display fireworks on the screen with a message that said "Happy 99", and then the virus would send itself to every email address in the infected computer's address book with the simple message Happy99. (McKinley, 1999) The Happy99 virus spread across the internet like wildfire and infected users all over the world. In many ways it was a harbinger for the next phase of malware.

The next phase in the evolution of malware is network worms. Network malware became increasingly popular as the internet became more wide spread. As the malware spread across networks, especially networks connected to through the internet, people started to realize the internet had a malware problem. One of the notable early viruses was the ILOVEYOU virus. This piece of malware hit computers around the world beginning on May 4, 2000. The malware was a simple virus that would infect the system and send a copy of itself to everyone in the infected user's address book. It is estimated the ILOVEYOU virus infected ten percent of the computers connected to the internet in a single day and close to 50 million systems in nine days. This was a shocking development in the world of information security. Another notable piece of malware from this era is the Code Red worm. Code Red was a worm that attacked Microsoft Internet Information Servers (IIS). It was estimated that 359,000 internet servers were infected using a buffer overflow attack. (Howard & Prince, 2011)

The fourth category is ransomware. Ransomware has become an especially prominent form of malware. Ransomware infects a computer and encrypts the user's file or locks the computer and then demands a ransom usually in the form of bitcoins in exchange for the key to decrypt the files or to unlock the computer. This malware has become popular as it provides profit for the malware authors. A recent report from Cisco stated, “When adversaries establish campaigns that compromise tens of thousands of users per day with little or no interruption, the ‘paycheck’ for their efforts can be staggering." The report also outlined a scheme by which hackers targeted 90,000 victims per day and netted an estimated $34 million annually in their operation." (Risen, 2016) The first of the true locking malware, Trojan.Randsom.C, attacked users around in 2008. This malware faked a Windows Security Center message and requested the victim call a premium-rate phone number to reactivate a license for security software. The malware locked the computer preventing the user from using the computer for any other purpose. (Savage, Coogan, & Lau, 2015) The first crypto ransomware, Trojan.Gpcoder, was released in 2005. Fortunately, the encryption was poorly implemented and was trivial to break. (Savage, Coogan, & Lau, 2015) However it did pioneer the way for other more sophisticated crypto ransomware malwares to be developed. Ransomware has been known to target home users, businesses, and government organizations usually with different amounts asked for based on the organization or person's ability to pay.

Finally, the latest category before fileless malware is the malware created by nation states and used for espionage and sabotage. Malware has changed from being a problem that businesses and private users face but also is viewed as a weapon wielded by Military and Spy agencies from various nation states. The United States government has determined that the military has the right to use and defend against cyber-attacks. One of the most widely known uses of malware to attack a physical installation is the Stuxnet virus. Stuxnet was discovered in June 2010. The malware targeted the Iranian nuclear facility at Natanz. Stuxnet is estimated to have infected over 60,000 computers, the majority of them in Iran. (Langer, 2011) German expert Ralph Lagner described Stuxnet as a military-grade cyber missile that was used to launch an ‘all-out cyber strike against the Iranian nuclear program.' (Farwell & Rohozinski, 2011) This was big step in the evolution of malware and one of the first times malware was used for the purpose of attacking another countries nuclear abilities. Another notable example of malware in this categories is Flame. Flame was found in 2012 and is believed to be the creation of Israeli and US secret services. The malware is light weight and modular with the ability to add on modules remotely. The malware has the ability to spread over the network or through USB devices. Flame installs a rootkit and allows the attacker to record audio, video, and skype calls. Also, Flame monitors network traffic and exfiltrate file from the victim. Another feature is the malware has a remote kill switch that will destroy the instances of Flame malware. The malware created by nation states tends to be complex and used for specific purposes. (Milošević, 2013) This category of malware is used for spying, sabotage, and other cyber-attacks against targets.

Fileless Malware

Throughout the history of malware one thing remained constant, someone had to create the code and develop the malware. Significant time and effort was put into programming the malware and working to evade anti-virus programs. In 2002, malware was about to enter into a new phase, the fileless malware phase.

.NET Framework

Microsoft released the framework, .NET, that changed the software development industry and unintentionally revolutionized the malware industry. The .NET framework provided malware writers with a new arsenal of weapons that could help spread malware and achieve the goals of the malware authors. The new approach quickly replaced the outdated formula of creating malware from scratch and working to stay ahead of the anti-virus companies. The new approach provided the malware authors easier and faster development from the nice framework that was provided in .NET. This framework made it harder for anti-virus programs to distinguish between malicious .NET activity and legitimate .NET activity. By using the .NET system, administrators and developers have the ability to interface with the Windows operating system, this also includes almost whole Microsoft product line, including Microsoft Office, SQL Server, Access, and so forth. This functionality is an incredible tool for developers but also makes malware development easier and gives the attackers the ability to easily develop and exploit vulnerabilities in the entire catalog of products developed in the .NET framework. (Pontiroli & Martinez, 2015) The value to a malware author is tremendous, an author can develop simple applications to send spam email, brute force credentials, or exploit vulnerabilities on a large scale.

Since much of this new malware is developed in the .NET framework, malware authors have started to take advantage of the integrated development environment (IDE) such as Visual Studio. Having access to a powerful IDE, allows the malware author to manage the lifecycle of the malware, update quickly, and rapidly develop malware with a consistent approach. The IDE also allows malware authors to setup and use effective test and development environments, which then allows them to sell their malware for more money since the quality of the product is better. (Pontiroli & Martinez, 2015)

PowerShell

While .NET has made malware development significantly easier, the holy grail of attacking has quickly become using PowerShell. In many ways, PowerShell offers tremendous flexibility and power for all stages of an attack and since it evades most anti-virus detection. Another added benefit of using PowerShell is that many times the modules used by the malware are whitelisted by the system administrators for legitimate use. However fileless malware really gets its reputation from the use of the PowerShell scripting language. PowerShell is tightly integrated into the Microsoft Windows environment and it is hard and impractical for many system administrators to turn it off. (Chickowski, 2016) The key to fileless malware is PowerShell. PowerShell scripts can be loaded into the memory of the operating system can execute commands without ever writing files to the hard drive. By dynamically loading the PowerShell scripts, the malware is able to attack the system without leaving any file based evidence. (Pontiroli & Martinez, 2015) This ability also provides the ability to hide from file based antivirus protection.

Another important ability of PowerShell is to interface with most of Microsoft products natively. This ability allows the malware author to increase the attacks that are available from a given malware. For example, PowerShell gives the attacker the ability to attack an Active Directory instance then use the information obtained to pivot and attack a SQL Server through the malware through the use of PowerShell. (Pontiroli & Martinez, 2015) These abilities give the malware flexibility and without incurring a significant increase in development cost.

Offensive PowerShell Tools

The use of PowerShell for offensive purposes, often called weaponized PowerShell, has been growing in popularity. Many of the Pen testing Kits, including the Metasploit Framework, contains PowerShell modules that are already built into the tools. While these pen testing tools allow offensive security engineers to look for vulnerabilities, they are also used by attackers for more nefarious purposes. Additionally there are other programs such as Mimikatz or PSattack that use weaponized PowerShell scripts to penetration test environments. (Metcalf, 2016) Examining various PowerShell offensive tools can show the many of the abilities used by fileless (Metcalf, 2016)Invoke Mimikatz is the PowerShell version of the popular Mimikatz offensive tool. Mimikatz gives attackers various post exploit capabilities. Mimikatz is primarily used after a victim's machine has been exploited and a foothold has been created. Mimikatz is infamous for allowing attackers to retrieve passwords from a Domain Controller. Other capabilities of Mimikatz includes retrieval of password hashes, pass the hash to other servers, and creation of golden Kerberos tickets. Passing the Hash and golden Kerberos tickets allow the attacker to move laterally through the network with fake or stolen credentials. (Mulder, 2016)

PSattack is a self-contained PowerShell console that contains many popular attack modules. PSattack is fileless in that the program can be loaded into memory and may never need to touch the disk drives. The payloads of PSattack come encrypted by default to evade antivirus programs. Also included in the program is an ability to rebuild the PSattack console with a custom encryption for evading antivirus programs. Most of the popular offensive PowerShell tools are included in PSattack. This list includes Powersploit, Invoke Mimikatz, Get GPPPassword, Invoke NinjaCopy, Invoke Shellcode, Invoke WMICommand, VolumeShadowCopyTools, PowerTools, PowerUp, Nishang, Powercat, and Inveigh. This collection of tools gives the user some flexibility and a tool for various stages in an attack from reconnaissance to exfiltration. (Metcalf, 2016)

PowerShell Forensic Tools

Another key to studying fileless malware is decompiling and reserve engineering the malware. Since much of the fileless malware is created in the .NET framework, the task of analyzing the malware easier. There are various free and open source decompilers that can help malware analysts understand the abilities and intentions of various fileless malware tools.

PowerShellArsenal, developed by Matthew Graeber, is .NET reverse engineering and Malware analysis tool. PowerShellArsenal was previously a standalone module inside the PowerSploit suite, but it has evolved into separate tool. Inside PowerShellArsenal is a collection of modules that offer various capabilities including performing memory analysis, collecting system information, and examining various file formats. Inside the programs the various tools are categorized with categories such as disassembly, memory tools, malware analysis, parsers, and Windows internals. The tool allows malware analyst to have all the tool they need to analyze various malware samples from a single tool set. This allows for faster and more consistent analysis of malware samples. (Pontiroli & Martinez, 2015)

Another important tool in the fight against fileless malware is an incident response framework. Kansa is a popular PowerShell based incident response network. The framework developed by Dave Hull automates the process of collecting key pieces of information during a security incident. The scripts can be executed locally or remotely. This tool allows you to customize the way the collected data is stored. The collected data coupled with a log parser can allow the incident responder to perform queries on the data collected. The ability to query logs aids the incident responder in understanding the security incident and to begin formulating a response. (Pontiroli & Martinez, 2015)

Conclusion

Fileless malware is the latest threat in the evolution of malware. Malware has evolved from simple malware that infected the early days of computers to the sophisticated fileless versions we see today. While similar to many of its predecessors, fileless malware is inherently different as it relies on capabilities found the operating system environment to accomplish its goals. The malware author's use of .NET and PowerShell will cause the information security industry to evolve in order to combat this latest threat. The future of information security will need to rely less on file signatures of malware since this new threat may never write a file to the hard drive. Instead, anti-virus programs will need to inspect the memory and evaluate how a program is being used. The key to the future of information security will be looking for behavior that is out of the ordinary and stopping malware that is using components of the operating system in a way that is not consistent with normal activity. In order for programs to have this capability research and money should be directed towards better understanding fileless malware and its behavior. Behavior based security is the future and as machine learning and artificial intelligence improve one of the key application needs to be information security. Another key to fighting fileless malware will be protections at the operating system level. Operating systems will need to adapt to this new threat and begin to implement better protections surrounding what is loaded and executed in memory. The operating systems have made some improvements towards this end but better validation and verification of programs should be implemented at the operating system level to protect the users from threats. While the evolution of malware has moved at a steady pace, the information security industry has also transitioned and adapted to the threats. In conclusion, while fileless malware looks to be unstoppable, new technologies and approaches to security will be developed and continue to keep users and their data safe.

References

Chickowski, E. (2016, December 27). *Fileless Malware Takes 2016 By Storm*. Retrieved from Dark Reading: http://www.darkreading.com/vulnerabilities---threats/fileless-malware-takes-2016-by-storm/d/d-id/1327796

Farwell, J. P., & Rohozinski, R. (2011). Stuxnet and the Future of Cyber War. *Survival*, 23-40.

Howard, D., & Prince, K. (2011). *Security 2020: Reduce Security Risks This Decade.* Hoboken, US: Wiley.

Langer, R. (2011). Stuxnet: Dissecting a Cyberwarfare Weapon. *IEEE Security & Privacy*, 49-51.

McKinley, D. (1999, May 25). Happy99 Newest Computer Adversay. *Wyoming Tribune*.

Metcalf, S. (2016). *PowerShell Security: PowerShell Attack Tools, Mitigation, & Detection*. Retrieved from Active Directory Security: https://adsecurity.org/?p=2921

Milošević, N. (2013). *History of Malware.*

Mulder, J. (2016). *Mimikatz Overview, Defenses and Detection.* The SANS Institute.

Pontiroli, S., & Martinez, F. R. (2015). The Tao of .NET and PowerShell Malware Analysis. *Virus Bulletin Conference.* Retrieved from Secure list.

Risen, T. (2016, July 27). *Ransomware Is the Most Profitable Hacker Scam Ever*. Retrieved from U.S. News and World Reports: https://www.usnews.com/news/articles/2016-07-27/cisco-reports-ransomware-is-the-most-profitable-malware-scam-ever

Savage, K., Coogan, P., & Lau, H. (2015). *The evolution of ransomware.* Symantec.