Anna TORCHYLO[1], Andrii BIGDAN[2]

Opiekun naukowy: Tetiana BABENKO[3]

ANALIZA „BEZPLIKOWEGO" ZŁOŚLIWEGO OPROGRAMOWANIA ORAZ MOŻLIWYCH ATAKÓW

Streszczenie: W artykule opisano główne zasady działania niezwykle popularnego bezplikowego szkodliwego oprogramowania. Ponadto odpowiada on na pytanie, czy nazwa „bezplikowa" może być właściwie używana dla tego typu programów i pomaga znaleźć sposoby skutecznej ochrony przed nim.

Słowa kluczowe: bezplikowe złośliwe oprogramowanie, niekonwencjonalne złośliwe oprogramowanie, złośliwe oprogramowanie w pamięci, ataki inne niż złośliwe oprogramowanie, wykrywanie złośliwego oprogramowania, luki w sieci

ANALYSIS OF "FILELESS" MALWARE AND ATTACKS

Summary: This article is describing the main principles of functionality of extremely popular fileless malware, that answers the question of whether the name „fileless" can be properly used for programs of this type and helps to find the ways of effective protection from it.

Keywords: Fileless Malware, Unconventional Malware, In-Memory Malware, Non-Malware Attacks, Malware Detection, Web Vulnerabilities

1. Introduction

The paradigm of global cyberspace cannot be differentiated from the concept of malicious software. The modern information space has become a new dimension for both economic and geopolitical confrontations, which gives a powerful impetus to the

[1] Taras Shevchenko National University of Kyiv, Faculty of Informational Technology, The Department of Cyber Security and Information Protection, speciality: Cybersecurity, atorouss@gmail.com

[2] Taras Shevchenko National University of Kyiv, Faculty of Informational Technology, The Department of Cyber Security and Information Protection, speciality: Cybersecurity, abigdan@gmail.com

[3] Doctor of Engineering Science, Professor, Taras Shevchenko National University of Kyiv, Faculty of Informational Technology, The Department of Cyber Security and Information Protection, babenkot@ua.fm

development of new, advanced forms of cyber threats as the main weapon of the invisible front – the information front.

The rapid development of modern multifunctional programming languages demonstrates the tendency to the emergence of new sophisticated malware families that take the threats of cyberspace to a higher level. This tendency can be demonstrated by the fact of the emergence of fileless malware, which has become one of the mainstreams of our time. This extraordinary popularity is understandable – fileless attacks are an example of Advanced Volatile Threats (AVTs), which are almost impossible to track, and at the same time, they are even more effective (in the level of damage they can cause to the system) than classic types of attacks.

This type of malware has been used since 1989. McAfee notes: "Frodo, Number of the Beast and The Dark Avenger" are the first examples of such kind of malware [1].

The denomination “Fileless Malware” speaks for itself. Performing fileless attacks is based on providing infection of the target system without requiring any additional files. It sounds almost like a science fiction because it is unclear where the infection comes from, how it works in the system, and reproduces itself. Therefore, this paper is made to identify the expediency of this title. Accordingly, we investigate the main features of fileless attacks in cyberspace.

The contribution of this paper is threefold. Initially, we identify the chef reasons for such a big popularity of this kind of malicious software. Moreover, the second part of our research is describing possible sources of fileless attacks. Finally, our research has aimed to find ways of protecting both advanced and ordinary users from harm that can be caused by malware of fileless type.

2. Basics of fileless malware

The main features can be understandable from the exploration of the malware’s life cycle.

In most cases, the initial infection with fileless software is quite trivial – through visiting malicious websites, opening phishing emails, malicious links, when users click on selected links, and download fileless payloads.

What's more interesting is that the fundamental difference from traditional viruses is that the code of the fileless malware is not stored in a file and is not installed on the victim's machine. It is loaded directly into random access memory (RAM) or other non-volatile storage components in the form of system commands [2]. This is the first and most important sign of fileless malware.

However, having conducted research about the features of these viruses, I dare to disagree with the previously given examples from McAfee. They should be called "predecessors of the genre" – the first stealth viruses that marked the beginning of the era of fileless attacks. Indeed, these programs were malicious RAM residents, but they required a file as a delivery system. Nowadays the term "fileless malware" has evolved significantly and adds much more to its meaning than only in-memory malware.

The initial payload file is a small embedded script, often obfuscated and partially encrypted to complicate internal monitoring. The task of this script is to get to the "inner sanctuary" and then run itself using the white list of Windows Script Host – wscript.exe or script.exe.

That is the first reason for the discrepancy between the name and essence of the program. In this case, the name of the fileless model may not be completely valid, because the installation is actually controlled by a separate program that includes a file.

The second important part of this process is that the attacker actually exploits existing vulnerabilities in software already installed on the computer – he or she does not need to download additional tools to the victim's machine. Instead, malicious software uses its own files and system services to give the attacker access to the device.

This principle is called LIVING-OFF-THE-LAND (LOL). In fact, there are more than 100 system tools used to implement this technique. This list (even three lists – LOLBins, LOLLibs, and LOLScripts) is kindly presented on GitHub [3], where anyone can use it for any purpose. Thus, the most powerful and popular tools, which are actively used by both regular users and administrators, can become powerful tools in attackers' hands within a few seconds [4].

Using LOLs seems like a win-win situation for attackers. LOLs reduce the number of files on disk and the number of actions that an attacker must perform for a successful attack, allowing him to gain a seamless but secure foothold on the system, working on behalf of legitimate system commands [55]. Fileless malware can execute routines according to instructions of the bot administrator, such as stealing information via formgrabbers, performing distributed denial-of-service (DDoS) attacks, updating itself, downloading and executing files, and accessing URLs.

The infection life cycle is shown in Fig. 1. It demonstrates the infection flow of fileless malware to simplify the understanding of the main steps of fileless attacks' mechanism.

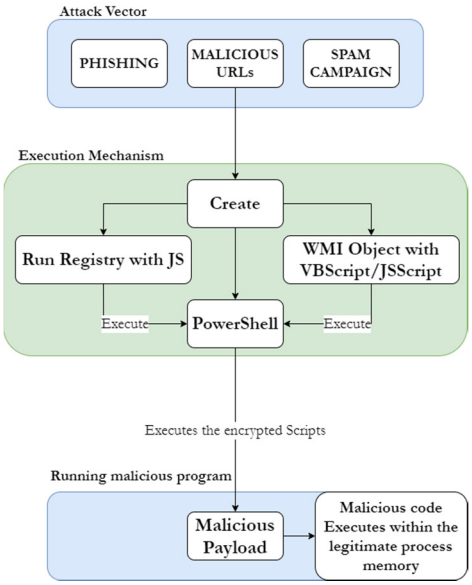

Figure 1. Infection flow of fileless malware

If you approach this philosophically, the attacker becomes a legitimate user of the system, that is why fileless attacks have received another name – non-malware attacks.

Because fileless malware launches its own operating system program instead of its own program, the actions of these instructions are not displayed in Task Manager under the guise of malware. Instead, everyone who checks for active processes will see the name of the interface that controlled the script's launch, such as PowerShell, and then see the common processes running under their own name [6].
This tactic is especially dangerous because the system and security technologies trust their tools, so an attacker can work secretly, for example, in the memory of a domain controller (a server that handles authentication requests) and even register system administrator credentials for deep penetration into the network, remote administration of infected hosts and collect privileged data.

2.1. Sources of penetration and execution

Most fileless malware attacks take advantage of the aforementioned MS PowerShell. Appealing to Symantec's 2019 report [7], 89% of the fileless malware detected in 2018 used this framework to hack the system.
PowerShell is a script interpreter. It allows system administrators to manage and automate tasks related to the running process, operating systems, and networks. PowerShell procedures are not blocked by firewalls or antivirus programs because they are ubiquitous in today's IT environments.
Another program that can be used for fileless attacks is the Windows Management Instrumentation (WMI). It provides information about the status of local or remote machines and can be used to configure security settings, such as system properties, user groups, scheduling processes, or disabling error logging [88].
WMI can be used to transfer commands to PowerShell. An example of a useful feature that WMI can perform for a hacker is the ability to activate WinRM (which controls PowerShell remote execution functions) if it has been disabled on the machine. WMI also gives a hacker access to the computer's registry [9].
Macros in Microsoft Office tools can also be used by hackers to deliver malware without files. They are usually created in Word documents or Excel spreadsheets as a set of commands grouped to run a task automatically [10], but malicious macros can also perform a variety of tasks, such as loading a malicious payload through PowerShell.
However, fileless malware is not limited to MS PowerShell. Attackers also widely use Java vulnerabilities to enter malicious code into target machines.
In particular, the CVE-2011-3544 vulnerability was exploited by virus writers in 2012, when more than 300,000 computers in Russia were infected with fileless software. This malware was distributed through news sites, namely advertisements placed on sites.
The attack took place according to the following algorithm:
Step 1: a user visits the infected site. No further action – he immediately finds himself on the server of cybercriminals.
Step 2: using the aforementioned Java vulnerability, the master server inserts an encrypted dynamic link library (DLL) file into the Java process (javaw.exe) on users' computers (which, of course, runs on RAM).
Step 3: information that includes technical details about the infected host is sent to the attacker's server.

Step 4: malware disables User Account Control (UAC). Malicious software then seizes the permissions required to install the target type of malware [1111].
In the case of Russian computers downloading malware, there was the Lurk Trojan, a program whose main function was to steal sensitive data for gaining access to Internet banking services. Later, Lurk was used in a fileless Java attack in 2016. The victim institutions lost 1.7 billion rubles. About 50 hackers were detained during the investigation of this attack [12].
JavaScript and HTML5 – a new generation of modern web applications – designed to increase the performance of web applications, their scale, and performance, they have serious potential for attackers [1313].
For example, we can use the method of creating interactive web applications Ajax (Asynchronous JavaScript and XML), which can open an additional page with a linked page in a pop-up window, hide, change the text in the status bar, change the text or graphics on the web-pages, create new cookies, modify or read existing ones. Since Ajax allows immediately to update the content of the web page when the user performs an action, so such malicious JavaScript code can be embedded in HTML and interpreted by a web browser transparently to the user.
Another common carrier of fileless malware is the Flash video playback system, which has a similar infection life cycle to JavaScript.

2.2. Fileless malware as a blind spot of the system

Traditional anti-virus programs are based on the principle of searching virus signatures – they scan hard drives, analyze files and, if they find some "traces" that may indicate the presence of a virus program, block it. At the same time, fileless software inhabits parts of the computer's architecture that are difficult for ordinary users to access – it exists only in RAM, which means that nothing is ever written directly to the hard drive.
In other words, fileless malware attacks place value on stealth, rather than persistence, though the flexibility of the attack to pair with other malware allows it to have both. Such kind of malware is a "blind spot" of ordinary antiviruses. That's why it is believed that fileless malware is almost impossible to be identified in the system (according to the Ponemon Institute, fileless attacks have a 10 times higher chance of success than file-based attacks). Moreover, this technology leaves little forensic evidence for security teams that can be used to investigate and reverse engineer a breach [14].
However, not everything is as bad as it may seem at the first sight. The main advantage of fileless software is its Achilles heel. Zero trace attacks (as they are also called) depend on the computer's RAM. This means that they can only work when the PC is turned on, which limits the ability to attack. Therefore, the most simple way to get rid of these types of threats is to restart your computer which arguably is problematic.
But if we look closer, in most cases, rebooting the system cannot be a panacea. The attackers are hooking into an API that guarantees that the process survives when the user closes the application. As far as system shutdowns go, most people will spend several hours on their computers before powering it down. This is long enough to capture keystrokes or perhaps even download additional malware. Fileless malware often has its own resilience scripts that are written to the operating system registry and restarted when it is turned on or scheduled. Normally, short lists of instructions do not

need to be stored in a file. However, longer and more complex scripts are saved for restarting during system startup [15]. This is the second reason why fileless software is not fileless indeed because it actually uses a file for resurrection.

3. True fileless malware. Is it real?

As we can see, fileless malware could mean tricking a user into running a script that executes a .NET binary directly from memory, like Sharpshooter which downloads the malware payload via the text records of DNS queries. Or it could mean sending malicious network packets that exploit the EternalBlue vulnerability and install the DoublePulsar backdoor in kernel memory. It could mean storing the malicious payload in the Registry as a handler for a file extension so it runs when you open a normal file with that extension. Kovter, for example, used that to download Mimikatz and steal credentials, putting the payload in a DLL that's encoded into a string and run with a PowerShell command, installing a malicious PowerShell comment in the WMI repository and configuring it to run at regular intervals. The malicious code could even be in device firmware or a peripheral like BadUSB; that way, the payload can run in memory and keep coming back even if you reboot, reinstall Windows or reformat the disk [166].
In previous sections of this paper, we have already talked about some of them. But, now we concluded, that they are not fileless software indeed. Anyway, there are really fileless malware in the list from Mary Branscombe, that was mentioned above.
To demystify the term, Microsoft Company have categorized fileless attacks based on how they get onto PC and where they are hosted. The datagram of this classification is shown in Fig. 2.

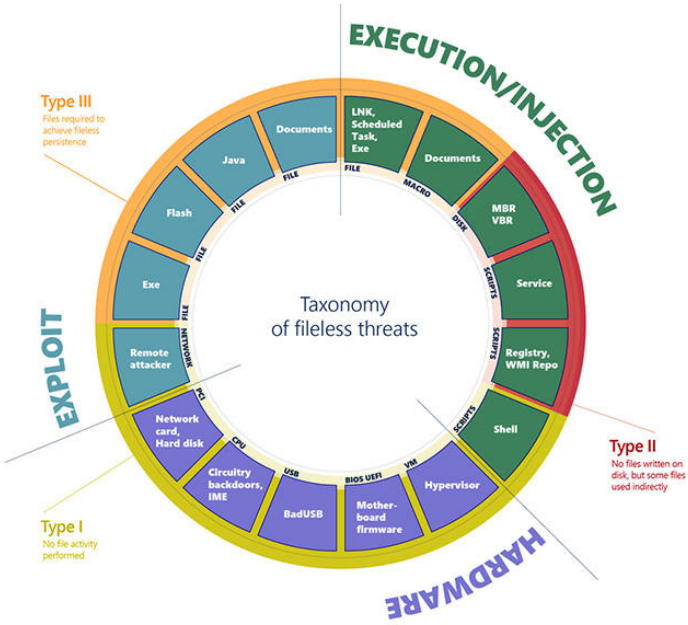

Figure 2. Categories of fileless attacks

There are more than a dozen combinations of those 'entry points' and malware hosts being used for fileless attacks. But they fall into three broad groups:

- Type I. No file activity performed
- Type II. Indirect file activity
- Type III. Files required to operate.

As we can see, type one is truly fileless, where the attack is delivered on the network or from a device, the payload is handled in memory and almost nothing touches the disk at all.

A fully fileless malware can be considered the only one that never requires writing a file on the disk. This type of fileless malware requires high levels of sophistication and often depends on particular hardware or software configuration. Threats of this type are uncommon and not practical for most attacks because they cannot be exploited easily. Anyway, they are real. And they are fully fileless.

How would such malware infect a machine? One example is where a target machine receives malicious network packets that exploit the EternalBlue vulnerability. The vulnerability allows the installation of the DoublePulsar backdoor, which ends up residing only in the kernel memory. In this case, there's no file or any data written on a file.

A compromised device may also have malicious code hiding in device firmware (such as a BIOS), a USB peripheral (like the BadUSB attack), or in the firmware of a network card. All these examples don't require a file on the disk to run, and can theoretically live only in memory. The malicious code would survive reboots, disk reformats, and OS reinstalls [177].

4. Preventive measures

To protect against fileless software, you need to stay one step ahead of it– intrusion prevention is better than a cure.

First of all, keep your software up to date. It can be hard to believe, but constantly updating the program and OS can highlight up to 85% of targeted attacks.

Microsoft, as the main victim of attackers, is very active in taking steps to block the exploitation of PowerShell and WMI, so, installing any updates from Windows Defender should be a priority. Windows Defender ATP's next-gen dynamic defenses have become of paramount importance in protecting customers from the increasingly sophisticated attacks that fileless malware exemplifies.

It provides the next principles:

Behavior Monitoring. Its engine filters suspicious API calls. Detection algorithms can then match dynamic behaviors that use particular sequences of APIs with specific parameters and block processes that expose known malicious behaviors.

Memory scanning. Even if malware can run without the use of a physical file, it does need to reside in memory in order to operate and is therefore detectable by means of memory scanning.

Boot Sector protection. It can be suitable for fileless threats because they can allow malware to reside outside of the file system and gain control of the machine before the operating system is loaded. And Windows Defender ATP does not allow to write operations to the boot sector [188].

At the same time, we must secure possible entry points. Fileless threats' attack vectors can include malicious sites and URLs, spam, and vulnerable third-party components like browser plug-ins.
Particularly dangerous are PDF files (you should turn off downloading PDF files in the browser and disable PDF readers from activating JavaScript.) and Microsoft Office macros. It is an extremely useful practice to disable unnecessary macro functionality.
Expanding Flash web video delivery systems can also be exploited. Most websites have already dismissed Flash and replaced it with HTML5 to include video. Therefore, blocking the system when surfing the Web will not be a problem.
As mentioned earlier, traditional antiviruses are powerless against zero-trace attacks. However, there are products that can resist them.
If the threat cannot be avoided, endpoint security should be installed in the system, which works not only on the basis of the analysis of malicious signals and analysis of user behavior. They can help monitor activity going in and out of the network, rather than just checking the files stored on it.
Such kind of programs allows us to solve the basics of blocking a web page that can host exploit kits, blocking the delivery of payloads if the exploit is detected in the system (before installing it), or by blocking the connection between your PC and malicious servers (if the utility has already been downloaded to the system due to Zero Day attacks).
Organization should use network- (NIDS) and host-based (HIDS) system intrusion detection, as well as endpoint analysis, to help determine indicators of compromise (IOC).
It is extremely important to proactively monitor the endpoints and networks. Fileless threats may not be as visible as other malware, but they can also leave telltale signs that IT and security teams could watch out for, such as suspicious network traffic [19].

5. Conclusion

Having conducted a detailed analysis of fileless software, we can say for sure that it is one of the best tools for a cyberattack. This family of malware allows getting maximum access, using a minimum number of resources, remaining invisible at the same time.
Nevertheless, fileless malware cannot be named fully fileless in most cases, because they are using additional files to perform attacks. In spite of this fact, ransomware propagation and infection are fileless. That is the only excuse to continue calling them in that way.
It is clear that this type of attacks has a huge potential and will repeatedly shake the world with its scale and audacity. And we have no choice but to evolve with them – to become more aware not to fall into the trap of intruders and, of course, to improve the security walls of our information systems that can withstand this level of fileless software sophistication to prevent and counter the spread of such malware.

REFERENCES

1. McAfee. – What is Fileless Malware: *https://www.mcafee.com/enterprise/ru-ru/security-awareness/ransomware/what-is-fileless-malware.html,* 2018.
2. Varonis – What is Fileless Malware? PowerShell Exploited: *https://www.varonis.com/blog/fileless-malware/*, 04.01.2020.
3. GitHub – Living Off The Land Binaries And Scripts: *https://github.com/api0cradle/LOLBAS*, 2018.
4. Search Windows Security – How do hackers use legitimate admin tools to compromise networks?: *https://searchsecurity.techtarget.com/answer/How-do-hackers-use-legitimate-admin-tools-to-compromise-networks,* 2018.
5. Minerva Labs – Deconstructing Fileless Attacks into 4 Underlying Techniques: *https://blog.minerva-labs.com/deconstructing-fileless-attacks-into-4-underlying-techniques*, 09.11.2018.
6. Security Affairs – Security experts at Trend Micro have discovered Phasebot malware, which also has fileless infection as part of its routine, is being sold online: *http://securityaffairs.co/wordpress/36206/cyber-crime/phasebot-fileless-malware.html,* 23.04.2015.
7. Symantec – Symantec's 2019 Internet Security Threat Report: *https://resource.elq.symantec.com/LP=6819?inid=symc_symc-home-page_ghp_to_leadgen_form_LP-6819_ISTR-2019-report-main&cid=70138000001Qv0PAAS&resetForm=1,* 2019.
8. Search Windows Server – Windows Management Instrumentation (WMI): *https://searchwindowsserver.techtarget.com/definition/Windows-Management-Instrumentation*, 2019.
9. Black Hat – Abusing Windows Management Instrumentation (WMI) to Build a Persistent, Asynchronous, and Fileless Backdoor: *https://www.blackhat.com/docs/us-15/materials/us-15-Graeber-Abusing-Windows-Management-Instrumentation-WMI-To-Build-A-Persistent%20Asynchronous-And-Fileless-Backdoor-wp.pdf*, 2015.
10. Cybereason – Fileless malware 101: understanding non-malware attacks: *https://www.cybereason.com/blog/fileless-malware,* 17.09.2019.
11. Search Enterprise Desktop – Java malware, fileless malware pose threats to desktop security: *https://searchenterprisedesktop.techtarget.com/tip/Java-malware-fileless-malware-pose-threats-to-desktop-security?_ga=2.192486826.1873762723.1600870900-1622005280.1599144934,* 04.12.2012.
12. Reuters – Russia says arrests hacker gang who defrauded banks of millions: *https://www.reuters.com/article/us-russia-cyber-arrests-idUSKCN0YN43L,* 01.06.2016.
13. SHERIF SAAD, FARHAN MAHMOOD, WILLIAM BRIGUGLIO. JSLess: A Tale of a Fileless JavaScript Memory-Resident Malware. Information Security Practice and Experience, 15th International Conference, ISPEC 2019, 25.11.2019.
14. Barkly – The 2017 State of Endpoint Security Risk: *https://cdn2.hubspot.net/hubfs/468115/Campaigns/2017-Ponemon-Report/2017-ponemon-report-key-findings.pdf,* 2017.

15. Digital Guardian – What is Fileless Malware (or a Non-Malware Attack)? Definition and Best Practices for Fileless Malware Protection: *https://digitalguardian.com/blog/what-fileless-malware-or-non-malware-attack-definition-and-best-practices-fileless-malware,* 12.09.2018.
16. TechRepublic – What is fileless malware and how do you protect against it?: *https://www.techrepublic.com/article/what-is-fileless-malware-and-how-do-you-protect-against-it/,* 11.09.2019.
17. Microsoft Docs – Fileless threats: *https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/fileless-threats,* 2020.
18. Microsoft Defender ATP Research Team – Out of sight but not invisible: Defeating fileless malware with behavior monitoring, AMSI, and next-gen AV: *https://www.microsoft.com/security/blog/2018/09/27/out-of-sight-but-not-invisible-defeating-fileless-malware-with-behavior-monitoring-amsi-and-next-gen-av/,* 2018.
19. AMIR AFIANIAN, SALMAN NIKSEFAT, BABAK SADEGHIYAN. Malware Dynamic Analysis Evasion Techniques: A Survey. ACM Trans. Web 9, 39, 2018.
20. Endpoint Security Solutions Review – Ransomware, Cryptojacking, and Fileless Malware: Which is Most Threatening?: *https://solutionsreview.com/endpoint-security/ransomware-cryptojacking-and-fileless-malware-which-is-most-threatening/,* 19.03.2019.
21. NortonLifeLock – What is fileless malware and how does it work?: *https://us.norton.com/internetsecurity-malware-what-is-fileless-malware..html*, 2019.
22. Malwarebytes Labs – Fileless Infections from Exploit Kit: An Overview: *https://blog.malwarebytes.com/threat-analysis/2014/09/fileless-infections-from-exploit-kit-an-overview/*, 14.10.2017.
23. Search Security – How does a new malware obfuscation technique use HTML5?: *https://searchsecurity.techtarget.com/answer/How-does-a-new-malware-obfuscation-technique-use-HTML5,* 2018.
24. Comparitech – Fileless malware attacks explained: *https://www.comparitech.com/blog/information-security/fileless-malware-attacks/,* 11.06.2018.
25. SUDHAKAR K., SUSHIL KUMAR. An emerging threat Fileless malware: a survey and research challenges. Cybersecurity 3, Article 1, 14.01.2020.
26. TrendMicro - Risks Under the Radar Understanding Fileless Threats: *https://www.trendmicro.com/vinfo/us/security/news/security-technology/risks-under-the-radar-understanding-fileless-threats,* 2019.
27. East Carolina University. David Patten – The evolution to fileless malware: *http://www.infosecwriters.com/Papers/DPatten_Fileless.pdf,* 2017.