Analysis of Fileless Malware and its Evasive Behavior

Asad Afreen
Internet Security & Quantum Technology Lab
NED University of Engineering & Technology
Karachi, Pakistan
arfeen@neduet.edu.pk

Moosa Aslam
Internet Security & Quantum Technology Lab
NED University of Engineering & Technology
Karachi, Pakistan
moosaaslam@neduet.edu.pk

Saad Ahmed
Internet Security & Quantum Technology Lab
NED University of Engineering & Technology
Karachi, Pakistan
S.ahmed@neduet.edu.pk

Abstract- Malware is any software that causes harm to the user information, computer systems or network. Modern computing and internet systems are facing increase in malware threats from the internet. It is observed that different malware follows the same patterns in their structure with minimal alterations. The type of threats has evolved, from file-based malware to fileless malware, such kind of threats are also known as Advance Volatile Threat (AVT). Fileless malware is complex and evasive, exploiting pre-installed trusted programs to infiltrate information with its malicious intent. Fileless malware is designed to run in system memory with a very small footprint, leaving no artifacts on physical hard drives. Traditional antivirus signatures and heuristic analysis are unable to detect this kind of malware due to its sophisticated and evasive nature. This paper provides information relating to detection, mitigation and analysis for such kind of threat.

Keywords- Advance Volatile Threat (AVT), fileless malware, evasion, malware analysis, executable malware, malware, Windows Management Instrumentation (WMI), PowerShell (PS), static and advanced malware analysis, memory analysis.

I. INTRODUCTION

Programs that are designed to change, damage and gain unattended access over the target system, they are generally known as malware. They are further categorized as viruses, trojans and worms. In the history of malware, Creeper is recognized as the first computer virus in 1971. To counter it, the reaper program was released, becoming the first antivirus [1]. In the early days of malware, authors spent hours writing assembly code for their malicious intent. The task of creating complex malware has become easier with the advent of higher level programming languages. In recent years there is a new trend in malware threat known as fileless malware. These malicious code written in such a wat that no anti-malware solution can detect it. Fileless malware are also known AVT [2]. These malicious programs are developed to install themselves in the system background on the target host computer. The objective of such malware may differ in terms of its malicious intent like collecting of personal data, banking information and system credentials or making the system a part of a botnet or even locking/encrypting personal data. When a system is compromised by the malicious program a forensic professional will look for malicious program or software that should not be on the target system, antivirus system known as anti-malware, designed to protect, detect and take action against malicious software. Antivirus solution starts by scanning files from the desired location and comparing the files on the disk against its signature database, however, in case of fileless malware attack there will be no file on the disk because the fileless malware does not reside on the physical drive and it is running in the memory. After writing malicious payload to the memory, hackers sometimes try to gain persistence by using legitimate system applications and tools such as Windows Management Instrumentation (WMI), Powershell (PS) and Windows registry. Fileless malware does not require a file to compromise the target host which is quite difficult to detect and prevent. A large percent of attacks involve methods of infiltration into secure networks, including fileless malware that can easily bypass security controls. Along with fileless malware, malware has many shapes and forms, such as computer bots, ransomware, worms, bugs, virus and trojans [3].

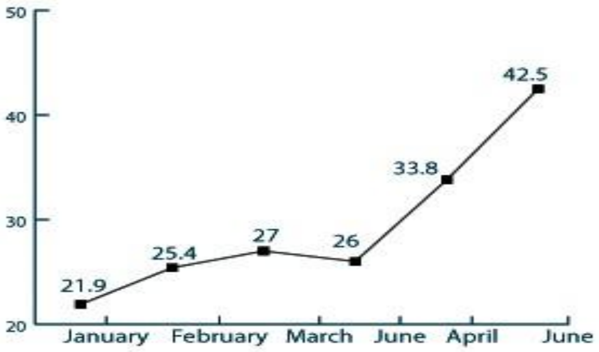


Fig. 1. Fileless attacks rose by 94% evidencing increasing prevalence, 42 attacks out of 1000 endpoints (Source:SentinelOne).

The main purpose of this paper is to study different forms of fileless malware, their infection techniques and legitimate system tools used in the process of infection, analysis of attack techniques of various fileless vectors and analysis of fileless malware.

II. LITERATURE REVIEW

This literature review give insights on malware and its types. Also working of antivirus, antimalware solutions and finally history of fileless malware along its working. Different types of malware like trojans, worms, ransomware, adware and backdoors are being studied in [4] including different propagation, detection and analysis techniques of traditional file-based malware. Most of the literature in our research that we came across was focused on file-based malware detection in which malware have digital footprint or techniques used for analysis of malware samples was file-based. Whereas our work compromises on methods of detection and mitigation for fileless malware because unlike traditional malware that uses malicious executables, fileless threats leverages trusted processes and applications. Methodologies being deployed by antivirus companies in order to overcome ever-growing threats of malware and outline of their strengths and weakness are being investigated in [5]. Detection of malware

978-1-7281-6840-1/20/$31.00 ©2020 IEEE©20XX IEEE

for antivirus solely depends upon there signature database and need to be updated frequently for reliable and timely malware detection. Antivirus, also known as anti-malware is designed to detect, prevent and take actions to block or remove malicious threats from computer system, some of the fileless malware detection techniques are presented in [6] in case of fileless malware it is much more difficult to detect as they are memory resident leaving no trace to detect them as the infection could be triggered with a malicious script or a benign executable used by computer system, Code Red and SQL Slammer (2001-2003) are earliest reference to fileless threats[7]. Fileless threats exploiting legitimate applications like PS, JavaScript and WMI are presented in [8] with different fileless threat attacks with comparison between file-based malware and fileless malware.

III. Methodology

Malware is an umbrella term used to refer to different forms and types of malicious software. Cybercriminals design malware to compromise computer functions, steal data, bypass access control and otherwise cause harm to the host computer [2]. Normally malware arrives via email, downloaded file or websites to allow the threat actors to install malicious program on target's machine, Fileless malware attacks are something where attackers uses techniques where malicious files are not written to the physical drive, they will use built-in legitimate tools such as PS or WMI to gain persistence and do network reconnaissance to know where they have landed in network. A fileless or AVT is a type of attack where the attacker takes advantage of pre-installed programs on the target machine. Unlikely, fileless attacks do need file-based malware to infiltrate target network. Fileless malware also leverages the legitimate process known as LOLBins (Living off the land Binaries). The attacker exploits the vulnerable application to reflect its malicious code directly into the main memory, Microsoft office, PS and WMI are used to run those scripts and load malicious payload [6] [9].

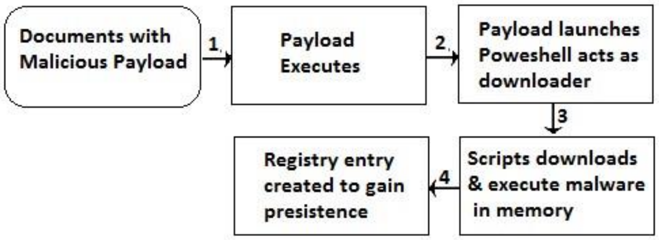

Documents with Malicious Payload
1
Payload Executes
2
Payload launches Poweshell acts as downloader
3
Scripts downloads & execute malware in memory
4
Registry entry created to gain presistence

Fig. 2. Typical document based fileless attack flow

Fileless Malware Access Paths

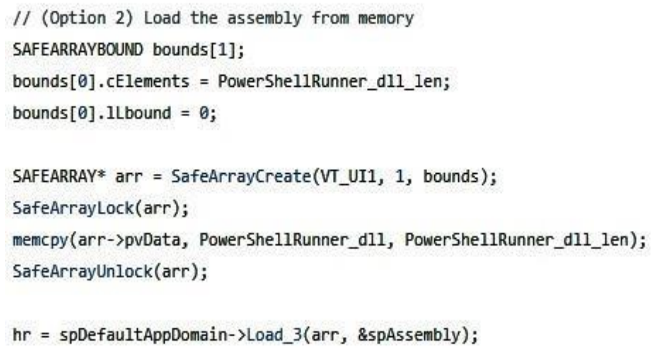
Threat actors still need to get access to target systems to achieve their malicious intent. The attackers would use tried and trusted methods to compromise their victim, it includes third-party application vulnerabilities and security misconfigurations. The percentage of PS and WMI-based attacks in 2016 reached 96% according to a threat report [10]. Access can be achieved by exploiting a remote code execution (RCE) vulnerability to run payload directly in memory. For example exploit in-memory (SMB EternalBlue, remote script execution) weak or stolen credentials (Remote Desktop Procedure, RDP) password. Another method of intrusion is an email with a malicious script embedded in the document or hidden in another host file (Email attachment with Non-PE file e.g. macro embedded document file). These kinds of files consist of multi-stage intrusion which may include downloader, dropper or self-decrypting parts and obfuscations. Dual-use tools especially tools like Mimikatz or Pwdump [11] [12] [13] and living off the land tools like macros, VB scripts, PS scripts, netsh, psexec are used. These kind of tools make it difficult for analyst and defenders to completely monitor and block access for attackers or to stop them from hiding in plain sight. Attackers using only pre-existing system tools and a handful of clean system applications are more than enough to carry out their malicious activities including stealing user information, data, credentials, allowing remote access. Preventing these types of attacks is an ongoing process, as the distinctions between normal use of these tools and malicious uses are very small.

Execution of Fileless Malware

Attackers have been using malware for its purposes to control the target system locally and remotely. Adversaries are taking advantage of vulnerabilities in the legitimate software programs that are already installed such as web browsers, Microsoft office, flash player and PDF viewers to exploit and load malicious payload directly into the main memory without accessing anything on physical drive. PS and WMI (Windows Management Instrumentation) are the two most powerful tools available on Microsoft Windows operating system in addition to these two programs .NET framework is also available which threat actors use to exploit vulnerability [14]. These tools have been maliciously used as a part of exploits. WMI is capable of system reconnaissance and is popular amongst threat actors. PS is a very powerful scripting platform and the shell is a highly flexible tool to evade antivirus and malware emulators and can be used for code execution, lateral movement, data theft and persistence.

A. . NET Framework

.NET frameworks comes preinstalled in Microsoft Windows operating systems to support software development tasks and other windows functionalities. Microsoft .NET framework comes in handy for malware authors, an emerging trend for threat actors for using .NET based AVT attacks to evade antivirus detection and mitigation, new approach provided malware authors easier and faster development of malicious code due to .NET ambiguous behaviour it is difficult to differentiate between legitimate and non-legitimate activity on the target system. Through the Assembly Load (byte [ ]) function (and its various overloads), the. NET framework has built-in functionality for dynamically loading memory-only modules. Enabling attackers to deliver fileless payloads to target systems.

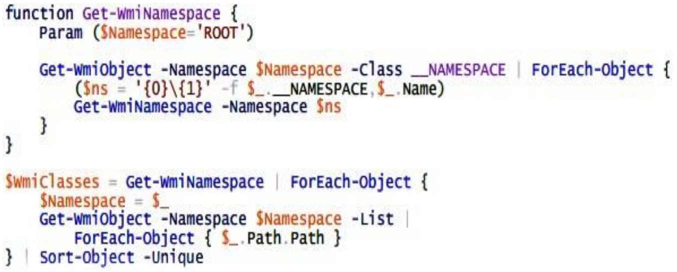


```
// (Option 2) Load the assembly from memory
SAFEARRAYBOUND bounds[1];
bounds[0].cElements = PowerShellRunner_dll_len;
bounds[0].lLbound = 0;

SAFEARRAY* arr = SafeArrayCreate(VT_UI1, 1, bounds);
SafeArrayLock(arr);
memcpy(arr->pvData, PowerShellRunner_dll, PowerShellRunner_dll_len);
SafeArrayUnlock(arr);

hr = spDefaultAppDomain->Load_3(arr, &spAssembly);
```

Fig. 3. Reflective Pick leverages the Assembly.Load() method to load their PS runner DLL without dropping it to disk [15].

B. Windows Management Instrumentation (WMI)

WMI is designed to provide system management information in a Windows operating system environment. Two percent of all malware submitted to Symantec sandbox in 2016 misused WMI [16]. Many changes have occurred since Windows 95 / NT over the time but WMI remains powerful, it is capable of launching attacks in many phases of the attack life-cycle like code execution, lateral movement and persistence. WMI can be used to execute malicious JavaScript/VB script payloads directly into the memory to evade traditional antivirus solutions which bases on signature detection [17]. WMI allows threat actors to interact with the target system using wmic.exe since it is a pre-built in windows capability, therefore it is difficult for antimalware/antivirus solutions to detect and restrict access from adversaries.

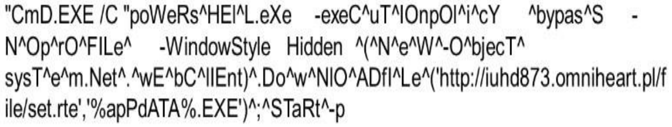


```
function Get-WmiNamespace {
    Param ($Namespace='ROOT')

    Get-WmiObject -Namespace $Namespace -Class __NAMESPACE | ForEach-Object {
        ($ns = '{0}\{1}' -f $_.__NAMESPACE,$_.Name)
        Get-WmiNamespace -Namespace $ns
    }
}

$WmiClasses = Get-WmiNamespace | ForEach-Object {
    $Namespace = $_
    Get-WmiObject -Namespace $Namespace -List |
        ForEach-Object { $_.Path.Path }
} | Sort-Object -Unique
```

Fig. 4. An example for PS code which can be used to recursively query all WMI classes and their respective namespaces [14].

C. PowerShell

PS is a very flexible and powerful program used by attackers. It can be utilize in many stages of an attack cycle, since it is a legitimate program pre-installed in Windows operating system and evades most antivirus detections. Payloads can be loaded into the memory of OS using PS and can execute instructions without writing anything to the disk, without leaving any evidence on disk. This ability makes fileless attacks based on PS more effective to hide from file-based malware protection solutions.



```
"CmD.EXE /C "poWeRs^HEl^L.eXe    -exeC^uT^IOnpOl^i^cY    ^bypas^S    -
N^Op^rO^FILe^    -WindowStyle  Hidden  ^(^N^e^W^-O^bjecT^
sysT^e^m.Net^.^wE^bC^lIEnt)^.Do^w^NlO^ADfl^Le^('http://iuhd873.omniheart.pl/f
ile/set.rte','%apPdATA%.EXE')^;^STaRt^-p
```

Fig. 5. macro malware payload that uses “^” for obfuscation calling PS [15].

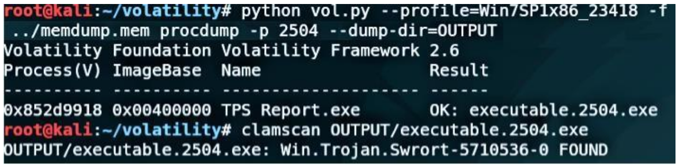
IV. Fileless Malware Attack techniques

A. Persistence Technique

Once the malicious code is loaded and executed possibly by starting the infection with a macro-based document. The attacker can misuse the utilities built into the operating system like PS to download additional malicious artifacts, launch programs and scripts, steal data, move laterally, and maintain persistence. After writing malicious content to memory, threat actors sometimes tries to gain persistence on the system. Also, adversaries often seeks control over legitimate user applications and system administration tools such as PS and WMI to execute and spread fileless malware. In fact, due to fileless behavior traditional antivirus solutions have failed to detect such threats, hence detecting fileless malware is challenging for security analysts. However, attackers can still exploit the vulnerabilities to steal data from a computer or even install other forms of backdoors to gain persistence, most fileless malware techniques are short-lived, and attackers use several evasive techniques like dual tool usage and Living off the land bins to achieve persistence. They do so by storing malicious code in unusual locations associated with the operating system or common utilities, such as the WMI Store, SQL tables,
Windows registry or tasks scheduling to inject malicious code into a system process, which helps threat actors to evade detection, as the activities would seem to be coming from legitimate processes. Payload directly passed as a command to CMD/PS and stored in OS scheduler task or windows registry, and executed by OS scheduler. Due to the application's nature to execute in the background as a service in most operating systems making fileless malware attacks succeed. Thus, fileless malware /AVT that execute in volatile memory (RAM) has more persistence when compared to other file-based malware. Attackers want to ensure persistence to exfiltrate data outside the network. Fileless persistence tactics to start their malware. There are various methods that an attacker can use for the fileless persistence attack procedure. This usually requires that the malicious code is already running on the compromised system in the post-infection phase.

B. Memory Resident Malware

Memory resident malware loads the payload directly into the memory-making detection and sign of infections difficult to identify. The malware continues to run after victim reboots the infected system, most of the memory-resident malware has been created in such a way that they inject code into registry entries. WMI events and background services will be later discussed in this paper. To detect memory-resident malware, antimalware solutions should be equipped with RAM captures to perform behavioral analysis on the dump to detect malicious processes and their child processes. Manual analysis of such type of attack could be done by using Volatility [18] and for memory capture FTK Imager [19] is a tool which can be use for this purpose. Once a memory dump is captured it can be analyzed in a controlled environment for analysis, Volatility Framework can be used for such kind of analysis.



```
root@kali:~/volatility# python vol.py --profile=Win7SP1x86_23418 -f
 ../memdump.mem procdump -p 2504 --dump-dir=OUTPUT
Volatility Foundation Volatility Framework 2.6
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x852d9918 0x00400000 TPS Report.exe       OK: executable.2504.exe
root@kali:~/volatility# clamscan OUTPUT/executable.2504.exe
OUTPUT/executable.2504.exe: Win.Trojan.Swrort-5710536-0 FOUND
```

Fig. 6. Process analyzed using volatility framework and detected using clamav signature database [20].

C. Windows Registry Malware

Windows Registry malware like Poweliks is different from of file-based malware due to its nature as a fileless threat. Poweliks persist by deeply embedding itself inside the Microsoft Windows registry. Windows Registry stores settings of the Windows operating system and some applications. In Windows Registry, the threat actors inject the complete malicious payload into the registry in an obfuscated manner, to make it evade detection. It exploits vulnerabilities to gain persistence. Due to its fileless nature, it destructs itself after completing its malicious intent without leaving any traces [16]. In windows registry malware persistence methods includes JavaScript code added into the registry and is executed by a legitimate Windows application, a PS script which decodes the encoded shellcode and injects it to the legitimate windows process to execute, using a technique called Process Hollowing which will be later discussed in this paper. Phase bot is a type of bot, which can grab its' victim information by applying a form-grabbing approach and stealing FTP data connection with the ability to run without a file. Phase bot hides its payload encrypted in the registry and uses PS to read and execute this independent position code into memory [23] [24].

D. Rootkits

This type of malware is kernel-based known as rootkits despite not being completely fileless, rootkits often reside at an OSI level below what antimalware solution can detect and can scan files to effectively evade detection. Signature-based antivirus solutions fails to detect these kinds of threats without extra protection modules like behavioral analysis. Attackers tries to completely remove their footprints, which leaves very little room for malware analysis. Rootkits are mostly installed on the target system to maintain a foothold. For lateral movement worms are placed within networks with rootkits, fileless behavior leaves the process of lateral movement in the network as a manual task [21]. Rootkits based attacks are sometimes classified as fileless attacks. By definition, fileless malware does not leave any trace on physical drive. Rootkits can access a computer remotely while staying undetected by security programs or administrator privileges. The rootkit can even alter the software that is specifically designed to detect rootkits [22]. Once the payload is injected into the target system adversary can infiltrate, gain access to data and sensitive information. After leveraging administrative privileges in windows operating system, Rootkits are installed in the post-exploitation phase of an attack.

E. Process Hollowing/Injection Attacks

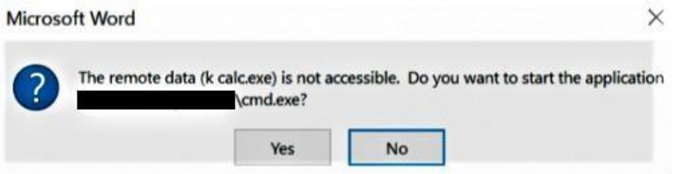
Process hollowing is a technique of hiding a process behind a legitimate process to hide in plain sight. The processes are simple, an application creates an empty/suspended process swapping out the original code from the process, then the legitimate process is injected with adversary malicious payload. Once the payload is loaded in memory, the process is then resumed with the entry point of the new payload which was injected. A technique somehow similar to process hollowing is Process injection, It is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, more sophisticated attacks may perform multiple process injections to further evade detection, system/network resources, and possibly elevated privileges. Payload injected using process injection may also evade detection from security products since the execution is executed under a legitimate process [25]. In order to detect such kind of attacks security companies should make a baseline in there antivirus solutions of operating system's processes which are mostly targeted, all other processes that are not included in the baseline and created during usage of operating system should be scanned and analyzed for any malicious indicators.

F. Reflective DLL Injection

Reflective DLL injection/loading refers to loading a portable executable (PE) from volatile memory (RAM) rather than from disk drive, Reflective DLL injection is an attack technique in which the concept of reflective programming is employed to perform the loading of a library from memory into a host process. Attackers insert malicious code into a legitimate process through DLL injection. DLL (Dynamic-link library) are the Microsoft's implementation of the shared library concept and provide a mechanism for shared code and data, a process referred simply as run-time dynamic linking by Microsoft, and its code is usually shared among all the processes that use the same DLL. Such files contain instructions that multiple programs can call a single DLL as needed during runtime. Using Legitimate Windows programs attackers conduct malicious injection without raising alerts. With reflective DLL injection, attackers are able to copy an entire DLL into process memory, which avoids having that DLL reside on disk (making it hard to detect) and being registered with the process it's being injected into [26]. A malicious payload can reflectively load portable executable without getting registered as module in the process and hence can perform actions without leaving forensics footprints [27].

G. Dynamic Data Exchange (DDE) Attacks

DDE stands for Dynamic Data Exchange (DDE) is a Microsoft Office feature that allows different office applications e.g. Word, Excel to exchange data between themselves by using shared memory. DDE is a client-server protocol for inter-process communication between Microsoft office applications. This feature has been updated into Office since Windows 2.0 in 1980's, but it is now being exploited by hackers to evade security detections. Dynamic Data Exchange attacks are the next generation of macro-based attack as they continue to use Office documents as an entry point into the target victim. The National Institute of Standards quoted "Users feel overwhelmed when they are confronted by countless security warning messages or have to be constantly vigilant. Security fatigue can cause users to act in ways that can put their computers at risk [27]. Traditional antimalware solutions will not detect these kinds of attacks and requests for command execution.



The remote data (k calc.exe) is not accessible. Do you want to start the application [redacted]\cmd.exe?

Fig. 7. Microsoft Word document prompting user to open cmd.exe [28].

H. Dual-use Tool Attacks

Dual-use tools are used by attackers for malicious intent but are also used by users if they have the required permissions. Due to major change in threat land escape adversaries are exploiting legitimate or dual-use tools like PS, netsh, PsExec.exe, etc. to deliver Memory only payload and Non-PE file payload. A tool that is used by attackers and system administrators, is Microsoft's PsExec. It is a command line tool that allows users to execute processes on remote machines and redirect output to the local system [29]. Dual use tools are sometime available on the target system and if not, they can be easily downloaded without raising any suspicion on the victim's machine. Attackers can take advantage of dual-use tools as these tools make it difficult for defenders to differentiate between malicious and non- malicious activity. By using commands, attackers are able to bypass most application security as well as some security checks. These tools are also misused for Reflective DLL attacks where a malicious DLL is dropped in the same directory as the legitimate one and is found first when Windows searches for the required DLL [26]. Dual-use tools are common in attacks due to their evasive nature which can bypass security controls [31].

V. Fileless malware Detection techniques

Fileless malware or AVT threats runs entirely in memory so as to avoid detection and leave very few footprints. New techniques have been developed in order to detect these kinds of threats. Behavioral analysis of a system is emerging technique in order to detect fileless threats by keeping tracking of previous activities. In order to protect your network and environment security professional should take pro-active approaches in detecting these kinds of threats. While fileless threats are hard to detect which causes delay in detection, incident response and memory forensics. Due its fileless nature it is becoming critical because fileless malware does not use traditional executables to compromise the target system, signature-based detection systems are not able to detect these kinds of threats because nothing is written to the file system if something is written in the early stages of fileless attack it would be erased in later stages of attack to maintain persistence.

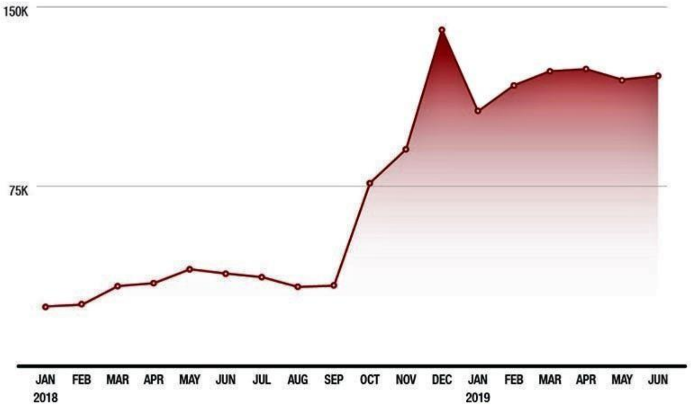


Fig. 8. Monthly fileless events (Trend Micro January 2018 to June 2019) [31].

A. Behavioral Analytics

Behavioral analytics is the feature which improve detection mechanism on endpoints with the help of artificial intelligence and machine learning which provides protection at runtime. Operating systems are still vulnerable to these kinds of attack where malicious payloads are directly loaded into the memory without raising any alerts. Behavioral Analytics of host operating system helps detect and stop fileless malware from ever being loaded into memory. Since fileless malware by definition has very little file-based indicators of compromise (IOC), analytics solution will record every kind of event that was attempting to load the detected malware into memory.

Security tools like anti-malware agents, intrusion detection/prevention systems (IDS/IPS), sandboxes, and firewalls must be installed with behavior analytics platform to prevent from threats and generate security alerts. If there is an alert which is anomalous, then a security analyst will be engaged in analysis of such alerts. Which could lead to malware artifices or a compromised endpoint. Behavioral analysis allows detection of AVT on early stages of attack. Behavior-based detections are based on analyzing execution patterns of any process in the system to detect any intrusion.

Since traditional detection programs use signatures to detect malicious file. Security companies have now focused on behavioral analytics to detect the malware. Some examples of abnormal behavior are unusual working hours, suspicious login activities, or accessing an unauthorized resource. This kind of behavior does not detect malware itself but it gives a starting point for security analyst to analyze the events, such kind of activities may lead to false positive at some point of investigation. Having behavioral analytics may help prevent data ex-filtration, information leakage and compromising of sensitive data.

B. Logging

In order to improve security posture and detection, organizations need to deploy and enhance logging, monitoring, and analysis of all network devices, endpoint, and user activity to identify fileless tactics. To enable such kind of monitoring organizations, have to deploy third party programs with capabilities such as endpoint protection and detection with memory and registry monitoring. Whereas PS logging must be updated to latest version, block unsigned PS scripts and execution policy. Limit PS to constrained language mode in order to limit PS to basic functionality which limits fileless attacks, enable PS extended logging for detailed logging, this kind of logging can result in a large number of EPS (event per second).

C. Least Privilege Rule

Another way to prevent fileless attacks is to implement principle of least privilege (POLP). Only the necessary rights should be assigned to the user that request access to a critical resources, giving user permissions beyond the scope can allow the user to obtain or change data and information in different ways. This method is not only limited to users but also to program, or process that should have minimum privileges necessary to perform their operations. This principle specifies some rules for example users account with least privileges, MySQL accounts with least privileges, just in time root privileges. Benefits of such kind of implementation leads to lesser threat levels to the network with better security and minimized attack surface.

D. Content Filtering

Content filtering depends on rules to filter out specific kind of traffic such as messages and their attachments e.g. Email security allows you to filter content based on your rules which consists of some attributes which identifies between malicious and non-malicious content. Another method is to filter web pages which host malicious content, this is generally achieved by web page filtering, implementing such kind of control may protect the network from drive by downloads in future. Content filtering rules applied using group policy such as Active Directory (AD) or LDAP and Layer 3/7 firewalls rules are used to filter out such kind of traffic within network.

E. Apply Patches and Updates

An outdated operating system is a vulnerable machine and is only waiting to be compromised. This can be overcome by rolling out patches automatically on time. Patching is important but not easy. Patching critical systems is a critical but an important task in order to improve security posture. Fileless malware threats are now present more than ever in order to protect your selves from such kind of threats which requires updated and non-vulnerable programs like keeping operating system and antivirus software with advance detection techniques intact and updated. Applying patches and updates can prevent most of the targeted attacks and the lack of patching is a major vulnerability, which results in most of the successful attacks on endpoints.

F. User Awareness

Companies invest a hefty amount on security controls in their network. A survey conducted in November 2018 asked 683 executives worldwide what percentage of their company's total IT budget was represented by IT security. The mean response was 15%. Nearly one quarter of the organizations, (23%) companies are devoting 20% or more of their IT budget to security [30]. In an IT environment, users are normally the weak point. They do not follow security practices, or bypass security measures for malicious intentions. However, user training and awareness can help educate users and raise awareness regarding cyber security and new types of attack vectors such as fileless malware, if the information system is very secure and isolate, but no user can use it and this can lead to security fatigue. If the system is configured towards only usability, then user with any privilege can access the system, this can result in a potential target for threat actors. For effective security training it should have some basic characteristics like, training must attract the attention of user, the message should be conveyed in simple as possible way to the user, training should be related to present security posture of company, training to end-users about attacks relating to fileless attacks, deception, social engineering and phishing.

VI. Fileless malware Detection Challenges

The theory of self-reproducing mechanism, which is the basis for most malware John von Neuman is credited. One of the newer concept of malware is fileless malware / AVT. Formal research papers are very limited in the field of fileless malware due to malware researchers that are in professional field, as they lack interest in publications of research papers. Whereas researchers in academia only relay on threat reports and other information related to fileless attacks published by security companies if available on internet. Therefore it seems a huge gap in between professional and academia research. Another challenge is the lack of malware samples obtained by researchers they only do their analysis on the few numbers of malware founds. Researchers obtain malware sample through compromised organization, samples published online, and honeypots to capture samples. Many forms of fileless malware yet to be analyzed due to insufficient malware samples. Some AVT are volatile and there artifacts gets lost after a system is reboot and the researchers are unable to analyzed such kind of malware samples. Antimalware solutions installed on endpoints to detect and protect against such kind of threats. In normal attack cycle for malware infection the endpoint is infected, exploited with a malicious payload and then executed. Mostly the entry point for such malware are emails, malicious websites/URLs and plugins. The common approach for antivirus solutions is to search for specific signatures that are available in signature database. Signature detection is not efficient because signatures that are previously created are made by capturing malware... hourly updates are common for signature databases and still there are gaps in between when a new malware is detected, analyzed and its signature is created. Endpoints are vulnerable during the time between a malware's analysis and a signature to be releases for antivirus. Security companies are now going to focus on minimizing damage after compromise. Heuristic analysis is another approach to detect malicious threat but is not sufficient to detect fileless threats. Heuristics refer to a set of instructions or rules to detect malware based on its behavior rather than on its payload. Heuristics analysis can be divided into static or dynamic analysis. Static analysis is the process of manually analysis of malicious code without actually executing. Dynamic analysis involves running the code and analyzing the results. The advantage to heuristic analysis is that it can detect unknown malware types. In case of fileless malware such kind of analysis may produce a huge number of false positive. Traditional antivirus search file system looking for malware signatures threat actors try to evade detection and obfuscate their code using encoding, packing, encryption and compression. Fileless attacks are increasing due to their evasive behavior because of very small footprint on physical drive which traditional antivirus programs are failed to detect. Dual use tools are also use in such kind of attacks; such tools are not flagged as malicious [32].

VII. CONCLUSION

Fileless malware or AVT is an emerging threat. Significant focus on detecting and preventing fileless malware attacks should be known and understood by all malware analysts because of its evasive behavior. This paper provides view regarding threats and emphasizes that there is so much work yet to be done in this field of fileless malware. Since fileless malware has very few file-based artifacts, it is very difficult to capture such kinds of threats. Since legitimate and dual use tools like PS, Macros and WMI are used in such type of attacks which can easily bypass signature-based detection solutions and make it difficult to detect exfiltration and malicious activities. In order to detect and interrupt fileless attacks, a multilayered approach is needed throughout the entire lifecycle of the fileless threat. Fileless malware will get more destructive with the availability of dual-use tools, which are readily available for malware authors. The purpose for this paper is to analyze why fileless malware is becoming more prominent, as well as difficult to detect and protect. Detection issues were discussed and techniques were proposed, which could be used by Malware Researchers towards making better detection mechanisms and sample collections for fileless malware.

REFERENCES

[1] A. P. Namanya, A. Cullen, I. U. Awan and J. P. Disso, "The World of Malware: An Overview," 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), Barcelona, 2018, pp. 420-427.

[2] Amir Afianian, Salman Niksefat, Babak Sadeghiyan, and David Baptiste. 2019. Malware Dynamic Analysis Evasion Techniques: A Survey. ACM Comput. Surv. 52, 6, Article 126 (November 2019)

[3] Dunst Consulting, "The different types of Malware Analysis," Medium, March(2017) .Available:https://medium.com/@dunstconsulting/the-different-types-of-malware-analysis-c9bfbaa44739.

[4] Eze, Aru Okereke. "Malware Analysis and Mitigation in Information Preservation." (2018) IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 4, Ver. I, PP 53-62.

[5] Babak Bashari Rad, Maslin Masrom, Suhaimi Ibrahim. "Evolution of Computer Virus Concealment and Anti-Virus Techniques: A Short Survey." (2018) International Journal of Computer Science Issues (IJCSI), Vol. 8, No. 1, 2011, pp. 113-121.

[6] B. N. Sanjay, D. C. Rakshith, R. B. Akash and D. V. V. Hegde, "An Approach to Detect Fileless Malware and Defend its Evasive mechanisms," 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions (CSITSS), Bengaluru, India, 2018, pp. 234-239.

[7] Lenny Zeltser. "The History of Fileless Malware – Looking Beyond the Buzzword." (2017) . Available: https://zeltser.com/fileless- malware-beyond-buzzword/

[8] K. S. Sudhakar, "An emerging threat Fileless malware: a survey and research challenges," SpringerOpen, Vols. Cybersecurity volume 3, Article number: 1, 2020.

[9] LOLBAS, "Living Off The Land Binaries and Scripts (and also Libraries)," LOLBAS. . Available: https://lolbas- project.github.io/.

[10] R. VALDEZ, "Who Needs Malware? PowerShell and WMI are Already There!," Carbon Black, 6 April 2016. . Available: https://www.carbonblack.com/2016/04/06/who-needs-malware-powershell-and-wmi-are-already-there/.

[11] Openwall, "Windows pwdump," Openwall, . Available: https://www.openwall.com/passwords/windows-pwdump.

[12] A. Greenberg, "He Perfected a Password-Hacking Tool—Then the Russians Came Calling," Wired, 11 September 2017. . Available: https://www.wired.com/story/how-mimikatz-became-go-to- hacker-tool/.

[13] B. DELPY, "Mimikatz," Mimikatz, 06 April 2014. . Available: https://github.com/gentilkiwi/mimikatz

[14] M. Graeber, "Abusing Windows Management," 2015. . Available: 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.

[15] J. Desimone, "Hunting For In-Memory .NET Attacks," Elastic, 10 October 2017. Available: https://www.elastic.co/blog/hunting-memory-net-attacks.

[16] C. Wueest and H. Anand,Internet Security Threat Repot (ISTR) "Living off the land and fileless attack techniques," Symantec, 2017. https://docs.broadcom.com/docs/istr-living-off-the-land-and-fileless-attack-techniques-en.

[17] F. Block and A. Dewald, "Windows Memory Forensics: Detecting (Un)Intentionally Hidden Injected Code by Examining Page Table Entries," Digital Investigation, vol. 29, no. Supplement, pp. S3-S12, 2019.

[18] M. Auty, A. Case, M. H. Ligh, J. Levy, A. Walters and J. Nick L. Petroni, "Volatility,"github, .Available:https://github.com/volatilityfoundation/volatility.

[19] Access Data, "Forensic Toolkit," Access Data . Available: https://accessdata.com/products-services/forensic-toolkit-ftk.

[20] Red Scan, "Memory Forensics: How To Detect And Analyse Memory-Resident Malware," Red Scan, 6 April 2018 . Available: https://www.redscan.com/news/memory-forensics-how-to-detect-and-analyse-memory-resident-malware/.

[21] J. Wilhelm and T.-c. Chiueh, "A Forced Sampled Execution Approach to Kernel Rootkit Identification," in Recent Advances in Intrusion Detection ,Springer, 2007, pp. 219-235.J. Wilhelm and T.-c. Chiueh, "A Forced Sampled Execution Approach to Kernel Rootkit Identification," in Recent Advances in Intrusion Detection , Springer, 2007, pp. 219-235.

[22] N. DuPaul, "Common Mobile Malware Types: Cybersecurity 101," VeraCode, 2 October 2013. . Available: https://www.veracode.com/blog/2013/10/common-mobile-malware-types-cybersecurity-101.

[23] Malware Tech, "Phase Bot – A Fileless Rootkit (Part 1)," Malware Tech, 11 December 2014. . Available: https://www.malwaretech.com/2014/12/phase-bot-fileless-rootki.html.

[24] Malware Tech, "Phase Bot – A Fileless Rootkit (Part 2)," Malware Tech, 14 December 2014. . Available: https://www.malwaretech.com/2014/12/phase-bot-fileless-rootkit-part-2.html.

[25] Mitre ,"Process Hollowing", Mitre 31 May 2017 . Available: https://attack.mitre.org/techniques/T1093/.

[26] The Barkly Endpoint Protection Platform, "The Fileless Attack SURVIVAL GUIDE," The Barkly Endpoint Protection.Available : https://dsimg.ubmus.net/envelope/395823/551993/Fileless%20Attack%20Survival%20Guide.pdf

[27] B. Stanton, M. Theofanos, S. Prettyman and S. Furman, "Security Fatigue" in IT Professional, vol. 18, no. 05, pp. 26-32, 2016

[28] Saif, "Macro-less Code Exec in MSWord," Sensepost , 2017 October 2017. Available: https://sensepost.com/blog/2017/macro-less- code-exec-in-msword/.

[29] M. Russinovich, "PsExec," 29 June 2016. . Available: https://docs.microsoft.com/en-us/sysinternals/downloads/psexec.

[30] B. Violino, "How much should you spend on security," csoonline, 20 August 2019. Available: https://www.csoonline.com/article/3432138/how-much-should-you-spend-on-security.html..

[31] TrendMicro, "Risks Under The Radar Understanding Fileless Threats," TrendMicro, 2019. Available: https://www.trendmicro.com/vinfo/us/security/news/security-technology/risks-under-the-radar-understanding-fileless-threats.

[32] Jesse Smelcer. "The rise of Fileless malware". in Partial Fulfillment of the Requirements for the Degree of Master of Science in Cybersecurity, December 2017.