19CSE311 Computer Security

Unmasking Fileless Malware: Advanced Memory Forensics with Volatility

Roll Number	Name
CB.EN.U4CSE22424	N Sai Kiran Varma
CB.EN.U4CSE22440	Soma Siva Pravallika
CB.EN.U4CSE22444	Suman Panigrahi
CB.EN.U4CSE22457	Sravani Oruganti

Abstract

The difficulties of fileless malware—an attack type that only uses volatile memory and uses trustworthy system tools to avoid detection—are examined in this case study. Fileless malware, in contrast to traditional malware, doesn't leave any disc footprint, so standard antivirus programs are useless against it. Investigators can detect unusual processes, code injections, and persistence mechanisms by using memory forensics, especially with the open-source tool Volatility, which allows them to record and examine temporary memory artefacts. Key attack strategies like process injection, registry-based persistence, and living off the land are described in the study, along with forensic investigation techniques like memory acquisition and thorough analysis workflows. There is also a discussion of workable mitigation techniques, ranging from regular patching and application whitelisting to limiting the use of scripting tools and improving logging. The outcomes highlight how crucial memory forensics is in modern cybersecurity defenses against evolving fileless malware threats [1]–[3].

Introduction

Fileless malware, a clandestine attack method that operates only in a system's memory without leaving traditional file-based traces, has grown in popularity among cybercriminals in recent years [1]. Unlike traditional malware that relies on executables written to disc, fileless malware uses legitimate system tools like PowerShell, Windows Management Instrumentation (WMI), and registry keys to execute malicious payloads. This technique, which is frequently referred to as Living off the Land (LotL) attacks, allows adversaries to avoid detection by conventional antivirus software [2].

Because fileless malware is transient, memory forensics has become an essential investigative technique. Investigators can retrieve evidence that would otherwise disappear when the system is shut down, including system logs, injected code, active processes, and network connections, by looking into volatile memory. One of the most popular programs for this purpose is the open-source framework Volatility, which provides a versatile, plugin-based method for removing digital artefacts from memory dumps [2], [8].

This case study explains forensic investigation techniques using Volatility, looks at the mechanisms of fileless malware attacks, and talks about mitigation strategies. This study reaffirms the importance of memory forensics in thwarting contemporary cyber threats by incorporating knowledge from current research and practical examples [3], [10].

Attack Techniques

Fileless malware exploits trusted system utilities to execute attacks without leaving traditional file traces, residing entirely in volatile memory [1]. The key techniques include:

1. Living off the Land (LotL) Attacks

Legitimate tools that are usually whitelisted in secure environments, like PowerShell, WMI, and application macros, are frequently used by fileless malware. Without setting off typical security alerts, these tools allow attackers to carry out reconnaissance, carry out payloads, and stay persistent [1], [10].

2. Entry Points and Initial Infection

To convince users to run these trusted tools, attackers commonly employ social engineering techniques like phishing emails or malicious links. Once running, the malware can use methods like code injection or process hollowing to insert malicious code into legitimate processes. It can also use registry changes or scheduled tasks to create persistence. For instance, the malware variant Kovter makes detection and removal difficult because it hides its payload inside registry keys and activates using PowerShell [1].

3. Execution and Persistence Mechanisms

Fileless malware uses several cutting-edge strategies to stay active:

- **Process Injection and Hollowing:** By introducing code into a trusted process, the malware hides its activities and exploits the host process's privileges.
- **Registry-Based Persistence:** Malware may store encrypted or obfuscated scripts in the Windows Registry. For example, Poweliks uses this method to re-execute after a system restart without being detected by standard disk-based scans.
- **Use of Scripting Languages:** An additional layer of secrecy is added by the fact that fileless malware can execute malicious commands in memory using scripting languages other than PowerShell, such as JavaScript or VBScript [2].

4. Real-World Example

In 2020, a fileless malware variant gained access to enterprise networks through a phishing campaign. The malware used PowerShell scripts to execute, process injection to move laterally, and registry changes to stay persistent. The challenge of identifying fileless threats without sophisticated memory forensic analysis is best illustrated by this multifaceted approach [3], [9].

Forensic Investigation Methods

Capturing and examining volatile memory is crucial to the forensic examination of fileless malware because conventional disk-based techniques are unable to identify these fleeting dangers. These crucial steps are typically followed in the process:

1. Memory Acquisition

The first step is to use programs like FTK Imager or DumpIt to take a precise snapshot of a system's volatile memory. Because fileless malware can only be found in RAM and capture errors can result in the loss of important evidence, a trustworthy memory dump is crucial [1], [7].

2. Analysis Using Volatility

Forensic investigators use Volatility to analyse the system's state at the time of capture after obtaining a memory dump. A variety of plugins from Volatility are available to extract valuable forensic artefacts:

- **Process Listing (`pslist`):** By listing every process that is currently active, this command enables investigators to spot unusual processes that might be signs of malicious activity.
- **Memory Artifact Detection (`malfind`):** This command assists in locating hidden malicious payloads that are stored in memory by looking for injected code and other anomalies.
- **Registry Analysis (`hivelist`):** Fileless malware's persistence mechanisms can be discovered by extracting registry hives from memory.

When combined, these tools offer a thorough picture of system activity, enabling investigators to identify fileless malware footprints [2], [8].

3. Forensic Workflow Overview

A typical forensic workflow using Volatility might proceed as follows:

- **Step 1:** Acquire the memory dump using a reliable memory acquisition tool.
- **Step 2:** Use `pslist` to generate a baseline of active processes.
- **Step 3:** Apply `malfind` to detect injected code or anomalies within these processes.
- **Step 4:** Extract and analyze registry information with `hivelist` to understand any persistence mechanisms.
- **Step 5:** Correlate findings with established fileless malware behaviors as documented in recent studies [7], [8].

Mitigation Strategies

A multi-layered strategy that incorporates proactive user training, ongoing monitoring, and technical controls is needed to mitigate fileless malware. An effective defence is outlined by the following tactics, which are backed by current research:

1. Restricting Scripting Tools and Execution Policies

Organisations should implement stringent execution policies, activate script block logging, and limit administrative privileges because fileless malware preys on trusted scripting environments like PowerShell and WMI. According to studies, properly configured PowerShell policies can drastically lower the frequency of fileless attacks [4].

2. Enhanced Monitoring and Logging

Early anomaly detection is made possible by robust monitoring, which includes thorough logging of command-line activity, process creation, and network connections. Faster incident response times are achieved by using SIEM systems to correlate these events [3], [9].

3. Application Whitelisting and Access Controls

The attack surface is reduced by restricting execution to authorised scripts and applications. Even if an attacker manages to get in, their ability to run malicious code will be severely limited thanks to the implementation of least privilege access. The effectiveness of whitelisting in stopping the execution of fileless malware payloads was shown in a case study from 2021 [5].

4. Regular Patching and System Updates

Updating software is essential. Frequent patching eliminates security holes that fileless malware could take advantage of. According to several reports, many of the initial infection vectors used in these attacks can be reduced by applying vendor patches on time [1].

5. User Education and Threat Intelligence

One of the most important lines of defense is user awareness. The initial compromise can be avoided by educating users about social engineering, phishing, and dubious email attachments. Organisations can also stay up to date on new tactics and adjust their defences by integrating threat intelligence feeds. Fileless malware attacks are less successful when user training and threat intelligence are combined [2].

6. Forensic Readiness and Incident Response

Rapid detection and remediation of fileless malware incidents is ensured by maintaining forensic readiness through frequent memory forensic drills and well-defined incident response plans. A real-world example from 2020 demonstrated how coordinated memory analysis helped contain an attack with little damage [3].

Challenges in Detection

Forensic investigators still face several inherent difficulties when attempting to detect fileless malware:

1. Ephemeral Nature of Evidence:

When a system is shut down, fileless malware leaves behind very few forensic artefacts because it only lives in volatile memory. Because of this fleeting existence, it is challenging to record and examine important evidence, which frequently leads to incomplete data sets during memory acquisition [1].

2. High Volume and Complexity of Memory Data:

Memory dumps contain enormous amounts of data that can be too much for conventional analysis techniques to handle. It takes a lot of computing power and experience to distinguish between malicious activity and legitimate process activity, especially when working with obfuscated or encrypted payloads [2], [7].

3. Obfuscation and Stealth Techniques:

Obfuscation techniques, like code injection into trusted processes and the use of legitimate scripting tools, are commonly used by fileless malware to conceal its presence. These strategies not only make detection more difficult, but they also make it more difficult for automated forensic tools to consistently spot unusual activity [2].

4. Integration of Legitimate System Tools:

The distinction between malicious and benign activity is muddled by the dependence on reliable system tools like PowerShell and WMI, which are essential to everyday operations. Because the same tools are frequently used for legitimate purposes, this integration makes it more difficult to create accurate detection baselines [1].

5. Dynamic and Evolving Attack Vectors:

Because fileless malware is constantly changing, detection techniques must also change. Because static detection models are unable to keep up with new strategies, ongoing research and dynamic analysis are essential to filling in the gaps in forensic techniques that currently exist [3].

Conclusion

As demonstrated by this case study, memory forensics is crucial for detecting and assessing fileless malware attacks. The inconspicuousness of fileless malware, which exploits reliable system tools such as PowerShell and WMI, renders conventional disk-based detection methods ineffective. Registry-based persistence, hidden process injections, and other malicious activities

are examples of temporary artifacts that investigators can discover in volatile memory by using forensic tools.

Along with the forensic investigation techniques used to record and examine memory dumps, key attack strategies employed by fileless malware have been described. Additionally, helpful mitigation techniques such as enhanced monitoring, regular system patching, scripting tool restriction, and application whitelisting have been discussed. The research has also highlighted intrinsic challenges in detection, including the ephemeral character of memory artifacts and the challenge of deciphering massive amounts of memory data.

All things considered, this analysis highlights how crucial memory forensics is as a component of the contemporary cybersecurity toolbox for thwarting fileless malware. Effective incident response and ensuring that organizations are resilient against new cyber threats depend on maintaining strong forensic readiness. [1]–[3], [7]–[10].

References

[1] "An emerging threat Fileless malware: a survey and research challenges," *Cybersecurity*, SpringerOpen. Available: https://cybersecurity.springeropen.com/articles/10.1186/s42400-019-0043-x

[2] "Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges," *Expert Systems with Applications*, Elsevier. Available: https://dl.acm.org/doi/10.1016/j.eswa.2022.119133

[3] "Resist Fileless Malware Threats," IEEE Computer Society. Available: https://www.computer.org/publications/tech-news/trends/resist-fileless-malware-threats/

[4] "Fileless Malware: What Mitigation Strategies Are Effective?" *BankInfoSecurity*. Available: https://www.bankinfosecurity.com/fileless-malware-what-mitigation-strategies-are-effective-a-11975

[5] "An Approach to Detect Fileless Malware and Defend its Evasive mechanisms," IEEE Conference Publication. Available: https://ieeexplore.ieee.org/document/8768769/

[6] W.-C. Chen, S.-S. Weng, and C.-H. Mao, "An Insight into the Machine-Learning-Based Fileless Malware Detection," *Sensors*, vol. 23, no. 2, p. 612, 2023. Available: https://www.mdpi.com/1424-8220/23/2/612

[7] "Effectiveness of Multiple Memory-Images in detecting Fileless Malware," IEEE, 2023. Available: https://ieeexplore.ieee.org/document/10131728

[8] "Near-Memory & In-Memory Detection of Fileless Malware," ACM, 2023. Available: https://dl.acm.org/doi/10.1145/3422575.3422775

[9] "Detecting Fileless Malware," LetsDefend Blog, 2024. Available: https://letsdefend.io/blog/detecting-fileless-malware

[10] "What Is Fileless Malware? Examples, Detection and Prevention," Fortinet, 2022. Available: https://www.fortinet.com/resources/cyberglossary/fileless-malware