Detecting Fileless Malware Attacks Through Windows Registry Analysis: A Cybersecurity Case Study

Anuja Chincholkar[1],*, Pallavi Shejwal[2], Anal Salshingikar[3], Smita Gumaste[1]
[1] Computer Science and Engineering Department, School of Computing, MIT ADT University, Pune, India
[2] Department of Information Technology, PCCOER, Pune, India
[3] MBA Cyber Security Management, National Forensic Sciences University, Gandhinagar, Gujarat, India
palhadeanuja@gmail.com, drpallavipatil@gmail.com, anal.salshingikar@nfscu.ac.in , smita.gumaste@mituniversity.edu.in

Abstract. The fileless malware has become a very elusive cyber threat, which uses genuine parts of the operating system to perform malicious functions directly in the memory without leaving the slightest trace in the form of a file. Such attacks are frequently not detected by traditional malware detection mechanisms that mostly use file signatures and static analysis. Windows Registry mechanisms have been used by at- tackers over the past years as a method of persistence, execution, and defense evasion, and Registry artifacts have become a critical but under- investigated source of forensic evidence. This paper includes a cyber- security case study that aims at identifying fileless malware by using systematic analysis of Windows Registry artifacts. The research paper reviews important Registry paths that are typically misused by the file- less malware such as start-up execution keys, Com hijacking entry, and environment-based persistence systems. The fileless attack scenario was controlled by use of legitimate windows utilities to monitor the types of Registry-level modifications that were created during the malicious execution. Registry artifacts obtained were examined to point out signs of compromise related to fileless behavior. The results prove that even though there are no malicious files present on the disk, fileless malware produces characteristic Registry-based signatures that can be used successfully to detect and examine it. The results indicate the applicability of Registry-based analysis as a light-weight and additional approach to other memory forensics and behavioral detection techniques. Registry analysis offers more insight into the persistence and execution patterns than traditional techniques, and requires less volatile memory acquisition. The research addresses a significant gap in the available literature about cybersecurity because it dwells upon the significance of Windows Registry artifacts in the process of identifying fileless malware. The pro- posed solution encourages better threat hunting, digital forensic investigations and incident response in the modern windows settings.

Keywords: Fileless malware · Windows registry · Registry forensics · Malware persistence · Memory-based attacks · Cyber threat detection · Digital forensic

M. Kaushik et al. (eds.), *Proceedings of the International Conference on Responsible, Risk-aware, and Regulated AI (RRRAI 2026)*, Advances in Intelligent Systems Research 210,
https://doi.org/10.2991/978-94-6239-723-1_37

1 Introduction

The rate of advancement of cyber threats has been so high that fileless malware is implemented, which is a well-structured type of attack and is accomplished without having a regular executable file. Fileless attacks do not remain on file systems as compared to conventional malware, but reside adjacent to legitimate parts of the operating system within system memory and as such cannot be easily detected by signature-based or file focused security services [2]. Consequently, the old methods of malware detection based on file system traces have become highly inefficient against these more covert attacker methods [1], [3]. Among the most significant points of fileless malware is a large-scale use of legitimate Windows components(PowerShell, Windows Management Instrumentation, and the Windows Registry) to maintain persistence, execute, and provide defense evasion [1], [6]. Of them, the Windows Registry has a central role, because the attacker can use the registry keys and values to sustain a presence even after re-booting the system, run some malicious scripts, and take normal control of the natural system operations, without leaving any traditional file based traces [16], [18]. Windows Registry contains configuration information needed in the running of windows operating system and installed programs. The fileless malware writers have been drawn to it, because it is trusted and highly intertwined with system processes [1], [7]. Registry keys like Run keys, Entries based on hijacking of a COM object, scheduled task entries and PowerShell execution policy are items also usually altered in fileless attacks and act as considerable sources of forensic evidence even though malicious executables are not created [25], [5]. The issue of identifying fileless malware by using a Windows Registry is a major challenge to cybersecurity experts, as well as, digital forensic investigators. As a result, there is an increased need to conduct dedicated research on identifying registry-based signifiers of compromise (IoCs) that can help expose fileless malware activity and conduct an investigation and response to it [13], [6]. This has resulted in a growing interest in devoted studies on the discovery of registry-based indicators of compromise (IoCs) that can divulge the presence of fileless malware activity and may be used to conduct productive investigation and response [19], [21]. Thus, in this work the analysis of is given attention. Windows Registry as a practical and efficient method of identifying and researching fileless malware intrusions. This study will help address this gap in the area of combating malware through traditional methods and the contemporary forms of malware applications, being memory-residents, by analyzing the registry-based indicators created during the implementation of fileless attacks, thus, improving the process of cybersecurity protection [24], [23].

2 Literature Survey

File-based detection systems fail against threats that execute wholly in mem- ory and abuse legitimate Windows components, so several authors have argued for a forensics-centric shift toward volatile and registry artifacts. Kara provides

a comprehensive overview of fileless attack techniques and emphasizes memory forensics as essential for detection and analysis [1]. Works on large-scale registry investigation and memory inspection show how registry modifications and in-memory indicators reveal persistent fileless activity that leaves little trace on disk (MemInspect; large-scale Windows registry forensics). These studies demonstrate that conventional signatures miss stealthy persistence mechanisms and that investigators must monitor volatile data sources and registry hives to reconstruct attack chains. [4][5][3] To improve detection, recent research combines memory-forensic feature engineering with ML and hybrid analysis pipelines. Sensor and deep-learning approaches extract behavioral and memory patterns to distinguish benign from malicious in-memory activity [2][3], while hybrid analysis models that fuse static, dynamic and volatile features show promise in catching obfuscated fileless payloads [8]. Image-based memory forensics and explainable- AI (e.g., texture descriptors + LIME) yield interpretable cues that help investigators localize malicious traces in memory snapshots [9]. Other studies of dark-web vectors and early-stage detection underscore the value of enriched threat intelligence and registry-focused indicators for timely response [6][7][10]. Together, these works justify developing registry-aware, memory-driven detection systems that integrate ML, explainability, and targeted registry feature sets to overcome the shortcomings of file-centric platforms. [2][8][9] Table 1 is a summary of the main research works concerning fileless malware infection published between 2020 and 2025. The analyzed articles are mostly related to memory forensics, behavioral analysis and machine learning-related detection methods to cope with fileless malware being evasive. Although these methods have been shown to be deferred detection and earlier discovery, the vast majority of the available literature offer minimal consideration to windows Registry-focused investigation. The analysis reveals that there is a definite research gap in utilizing Registry artifacts as the main signatures in the detection of persistence and execution of fileless malware, thus contributing to the rationale of a Registry-based case study of cybersecurity as the contents of the proposed paper work.

3 Problem Statement

Conventional malware detection platforms are generally based on file indicators, and thus ineffective in fileless malware attacks, which run directly in memory and abuse the legitimate windows components. Fileless malware is commonly persistent, being conducted using Windows Registry functions and defense evasion. This poses a great challenge to cybersecurity specialists because such attacks cause only a slight impact on the file system and alter the important artifacts in the Registry. Consequently, windows registry based indicators need to be analyzed to be able to detect and investigate fileless malware attacks.

Table 1. Summary of Related Work on Fileless Malware Detection

Ref. No.	Authors & Year	Paper Title	Methodology Used	Key Contribution	Research Gap
1	Kara (2022)	Fileless Malware Threats: Recent Advances, Analysis Approach through Memory Forensics and Research Challenges	Survey-based analysis, memory forensics	Provides a comprehensive survey of fileless malware techniques and forensic challenges	Limited focus on Windows Registry–specific indica- tors
2	Singh & Tripathy (2025)	Unveiling the Veiled: Early Stage Detection of Fileless Mal- ware	Behavioral analysis, system activity monitoring	Proposes early-stage detection mechanisms for fileless malware	Registry-based persistence artifacts not deeply explored
3	Zulkarnain et al. (2025)	ML-Driven Detection of Fileless Malware in Memory- Resident Atta ck Vectors	Machine learning on volatile memory data	Demonstrates effectiveness of ML in detecting memory-resident malware	Registry artifacts are not incorporated as detection features
4	Yaseen et al. (2025)	Image-Based Memory Forensics for Fileless Malware Detection	Deep learning, memory image analysis	Introduces image-based memory forensic techniques for fileless malware	High computational overhead; registry evidence ignored
5	Khalid et al. (2023)	Machine-Learning Based Fileless Malware Detection	Supervised ML models, memory behavior analysis	Evaluates ML classifiers for fileless malware detection	Lack of regi centric stry- extraction feat ure
6	Sherazi & Qureshi (2025)	Hybrid Analysis Model for Detecting Fileless Malware	Hybrid static and dynamic analysis	Improves detection accuracy using combined analysis methods	Registry persistence Mechanisms minimally discussed
7	Sanjana & Pavan (2024)	Detection of Fileless Malware from Dark Web	Behavioral anomaly detection	Explores stealthy attack techniques and behavioral indicators	Does not analyze Windows Re gistry modifications
8	Mutua (2023)	Advanced Analysis of Fileless Malware through Memory Forensics	Memory forensic framework	Proposes a forensic investigation model for fileless malware	Registry-based evidence not systematically studied
9	Dewan & Venu (2024)	Detecting and Investigating Fileless Mal- ware: A Deep Dive	Detection and investigation survey	Summarizes modern detection and forensic approaches	Lack of experimental registry-based
10	Khushali (2020)	A Review on File- less Malware Analy- sis Techniques	Narrative literature review	Provides foundational understanding of fileless malware analysis	Outdated techniques; lacks recent registry- focused approaches

4 Objectives

The main aims of the research are to investigate how fileless malware attacks behave in Windows environment, what Registry keys are often abused by mal- ware, how the power shell based fileless attacks execute and how to extract Registry-based Indicators of Compromise (IoCs), and demonstrates the effective detection strategies using a controlled case study of cybersecurity in this paper. The proposed research will examine the data artifacts, such as Registry items, PowerShell invocation logs, and persistence behaviors, and the study will ex- amine the behavior of fileless malware attacks in Windows environments. This paper focuses on finding regularly accessed Registry keys, drawing Registry keys Indicators of Compromise (IoCs), and learning how fileless dangers use legal opportunities in the system to be covert and endure. Through a controlled case study, the study shows how Registry analysis and forensic monitoring can be effective in detecting and characterizing fileless malware activity, which can be used to enhance the detection strategies and capabilities of response to incident in current cybersecurity activities.

5 Windows Registry Fileless Malware

The Windows Registry is a centralized hierarchical database that stores con- figuration settings, system policies, and execution parameters for the Windows operating system and installed applications [7], [23]. The Registry is vital to the start up of system and running of processes and has thus made it an easy target of fileless attacks. malicious software, persistence, stealth [1], [6]. Unlike conventional malware which is grounded on malicious binaries, Malware programs fileless steals legit registry keys (Run, Run Once, Services, COM class identifiers etc). Image File Execution Options (IFEO) and (CLSID) are turned on to execute malicious scripts or commands immediately. memory [16], [17], [23]. Due to these tricks, attackers can remain having extended access without notice. OS sig- nature based security devices identification [2], [8]. Windows Registry Windows Registry is a store of settings which is centrally located in a systematic manner. and is hierarchical which provides configuration setting, operating policy and system execution parameter of windows. applications and operating system in- stalled in it [7], [23]. Because of its central part in the starting of the system and the running of processes, the Registry has been targeted by fileless malware in pursuit of persistence and stealth [1], [6]. Fileless malware also uses legitimate Registry keys, i.e., Run, Run Once, Services, COM class identifiers (CLSID), and Image File Execution Options (IFEO), to execute malicious scripts or commands directly in memory as opposed to malicious binaries [14], [16], [18]. Such methods allow the attackers to have a long-term access and not be detected by the signature-based security tools [2], [15]. Registry modifications can be used as the main forensic footprint in the case of fileless attacks [1], [4]. There are also common reflection DLL injections, malicious PowerShell code, interpretable and encoded scripts that are generally initiated by the Registry-based executions [6],

[10], [16]. Also, countermeasures include Registry- based obfuscation and defense evasion tactics, such as concealed key, changed timestamps and use of trusted windows components by attackers [12], [18], [22]. Even though the artefacts that such attacks leave in the file system are little, they always alter Registry values to obtain execution and persistence, and Registry analysis is an effective method of post-incident analysis and threat hunting [5], [21], [23]. The case study aims at

Table 2. Key Windows Registry Locations Used in Fileless Malware Techniques

Registry Key	Purpose	Malicious Usage
HKCU\Software\Microsoft\Windows\CurrentVersion\Run	Startup execution	Persistence via PowerShell scripts
HKLM\Software\Classes\CLSID	COM object loading	COM hijacking for stealth execution
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options	Process behavior control	Debugger-based execution hijack
HKCU\Environment	User environment variables	Hidden script execution at logon

examining Windows Registry artifacts that occur in the simulated fileless malware activity with a view to finding credible evidence of compromise. Through scrutiny of modifications on keys associated with persistence, execution traces, and irregular Registry value distribution, the research shows the effectiveness of Registry-focused research in identifying covert malicious actions. The results emphasize that the systematic Registry analysis may supplement memory forensics and behavioral monitoring and offer an inexpensive though efficient way of identifying fileless malware in real-life conditions. The practice assists cybersecurity professionals in optimizing detection and boosting incident response measures against threats that are memory-resident and state-of-the-art.

6 Case Study Analysis

Both the current literature on fileless malware detection and file detection solutions mainly depend on memory forensics, machine learning, and hybrid analysis methods. The memory-based techniques can be efficient in detecting malicious code that is executed in the run time, but they have high overhead as well as minimal applicability in the post incident investigation [4], [5]. Machine- learned techniques increase diagnosis accuracy with volatile memory characteristics; nonetheless, they are regularly neither interpretable nor give substantial understanding of persistence phenomena [2], [3], [9]. Hybrid models also strive to integrate more than two indicators, yet they still rely on controlled execution backdrops [24], [10]. In comparison, registry-based studies underscore the windows registry as a stable and consistent artifact of the forensic file presence used in

the detection of fileless malware residency and deforcing malware. Alterations of Run keys, COM objects and execution policies are good pointers of compromise even without traces of file-systems [1], [6], [16], [25]. Nonetheless, the current literature on registries is not scal-oriented. capable and smart analysis frameworks, which disclose an information void that drives the registry based research in this research project [7], [19], [21].

7 Attack Simulation (SAFE ACADEMIC)

This is demonstrated by proving that persistence can be achieved without the malicious execution files being generated by executing fileless attack behavior using PowerShell with simulated behavior. to be executed when the system (or a user logging in) is started to cause commands whose memory-residence is re- quired, and to be file-system independent so that it can survive a reboot. The system perseverance is portrayed as follows in the powershell run. Installing a command to automatically run when a system is initiated or a user joins into the system. This application shows that power shell can be applied to attain the perpetuation of contact without producing executable programs hence recreating fileless perseverance conduct by lawful methods of system procedures. It is important to state that nothing is installed that is authentic malware that is created and deployed during the demonstration. It is an educational objective alone- to get understand the tricks of an assailant and to make it clear that record artifacts may be utilized as a concrete means of forensic evidence which may be employed to determine and investigate fileless remnants of persistence in case a security examination is performed. the figure. 1 illustrates the desired PowerShell-based simulation introduced in the case study as exhibiting persistence behavior, registry-based. The script directly executes in the memory and changes a known windows Registry start point so that it will automatically execute upon a user logging in the system. This controlled demonstration does not create or deploy any malicious executable files and is intended solely for educational and analytical purposes to understand fileless persistence techniques. Example Attack Actions: Registry-Based Persistence Simulation Using PowerShell This work demonstrates a simulated PowerShell-based fileless persistence technique using the Windows Registry. The objective of this simulation is educational and analytical, aimed at understanding how persistence can be achieved without creating any malicious executable files. No real malware is developed or deployed during this experiment. A typical, clear Registry state, i.e. a Registry state without any suspicious or malicious startup programs under the path of HKCU\Software\Microsoft\Windows\CurrentVersion\Run, is illustrated by Figure 2. A legitimate application usually uses this registry location to automatically start processes on user log on. This base has to be formed to compare it. HKCU\Software\Microsoft\Windows\CurrentVersion\Run As Figure 3 shows, the change in the registry was made in the course of the simulation whereby a new item by the name of \text{Demo\PowerPoint\Persistence} is added under the same key of run. The PowerShell script runs on-memorial, and verifies the existence of a

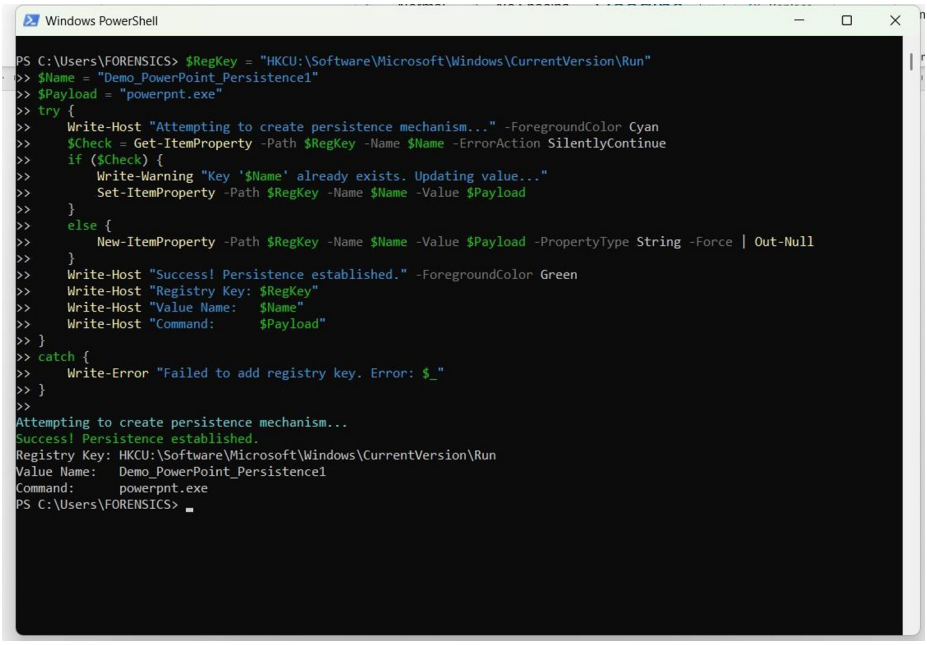

Fig. 1. Proposed PowerShell-Based Simulation of Registry Persistence

registry value. In case the value exists, then it is updated otherwise a new entry is made in the registry. The payload is targeting a legitimate Windows application (powerpnt.exe), that persistence can be maintained not only through reliance on external executable programs but also by using high-confidence sections of the system.

HKCU\Software\Microsoft\Windows\CurrentVersion\Run, This simulation emphasizes the creation of fileless persistence as possible by use of registry-based execution systems, where commands can be automatically invoked at an user login and file-system artifacts can be avoided. This kind of registry records does not disappear during system rebooting, and thus would be useful as a forensic indicator when investigating a post-incident and hunting threats. The experiment supports the emphasized usage of checking startup registry keys to identify stealthy persistence to fileless attacks techniques.

8 Results Discussion

Registry Keys Modified This demonstrates the process of modifying some of the keys in Windows Registry as it often would be the case in an attempt to create persistence in an attack with no file. The default registry key that was trailed by the simulation is: HKCU/Software/Microsoft/Windows/Current version/run in which a new entry was added enabling it to automatically start as soon as

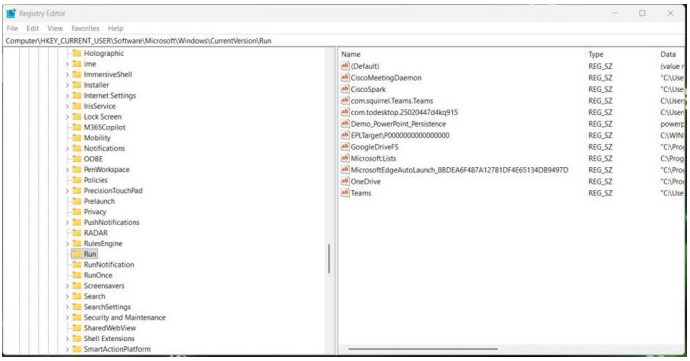

Fig. 2. Proposed Registry with Fileless Malware Execution which created

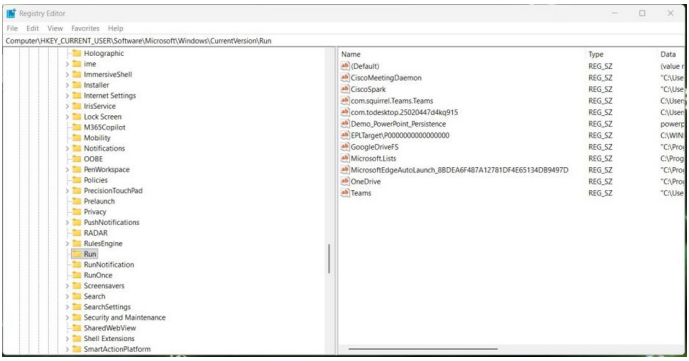

Fig. 3. Proposed Normal Registry Clean Entry

any user logs in. The move to switch this esteemed location of the start up the startup is one of the illustrations that perseverance can be executed with- out the system being loaded with any malicious executable files. Mechanisms of Persistence Viewed. The mechanism of persistence offered in this simulation is an execution of the system by means of registry at start up or access by a user. There are also the elements of the use of real windows-built elements, the best example of registry paths that bring out the ability of actions of memory to be remembered during the rebooting process. This confirms that persistence can be maintained even when file-system artifacts are minimal or absent. Com- parison: Clean vs Compromised System A comparative analysis was conducted between a clean registry state and the modified registry state after simulation. In the clean system, no unauthorized startup entries were present under the Run key. In contrast, the compromised state shows an additional registry en- try responsible for automatic execution. This comparison clearly distinguishes normal system behavior from persistence-enabled behavior and helps identify registry-based indicators of compromise.

9 Limitations

Even though one can make an argument that the study shows viability of the Windows Registry analysis as a tool to identify fileless malware, there are limitations to the study. It is an approach that is primarily composed of the observable Registry artifacts that can be altered, obscured or even deleted by more advanced offenders with the assistance of anti-forensic procedures. In addition, the case study was conducted under controlled environments and a small sample of fileless attacks, which may not be adequate to represent the occurrence and heinousness of real-life threats. The analysis does not incorporate the notion of real-time tracking, and extensive enterprise datasets and so, its scalability and applicability can be limited. Further, the plan focuses on windows based systems and does not look at fileless cross platform attacks. It implies that these limitations imply the need to emphasize the broader datasets, on-the-fly detection, and integrating memory and network forensics in future research.

10 Conclusion Future Work

Fileless malware can just operate inside the memory without leaving any or extremely minimal trace on the hard disk and is non-existent in traditional file based antivirus software. The research paper has demonstrated that to use windows registries artifact analysis can be helpful in determining the signs of persistence, execution and evasion by fileless threats. The results have shown that Registry-based investigation applied together with memory and behavioral analysis would enhance the accuracy and even greater depth of malware recognition and forensic examination. Future work will involve the development of ML-based Registry anomaly detecting models that are capable of detecting unusual alterations that include fileless malware automatically. Additionally the

real time tracking of Registry can be introduced that will also allow detecting the presence of stealthy behavior at the initial steps. The integrated use of these tools along with SIEM/SOC programs can also be used to scale the research, making it easier to alert threats centrally and/or automatically and speedily respond to the incidence in the corporate context.

11 Result Statement

The study revealed that even though there were no malevolent files, the research revealed that there are various Registry additions that can be linked to executions and persistence of PowerShell that have been found and that were a testament of the presence of fileless malware attack.

References

1. Khushali, V. (2020). A review on fileless malware analysis techniques. International Journal of Engineering Research and Technology, 9(6).
https://www.ijert.org/a-review-on-fileless-malware-analysis-techniques

2. Sudhakar, M., & Kumar, S. (2020). An emerging threat fileless malware: A survey and research challenges. Cybersecurity, 3(1).
https://doi.org/10.1186/s42400-019-0043-x

3. Tang, F., et al. (2020). RansomSpector: An introspection-based approach to detect crypto ransomware.Computers & Security, 97, 101997.
https://doi.org/10.1016/j.cose.2020.101997

4. Beaman, C., et al. (2021). Ransomware: Recent advances, analysis, challenges and future research directions. Computers & Security, 111, 102490.
https://doi.org/10.1016/j.cose.2021.102490
5. Ali, M., Shiaeles, N., Clarke, N., & Kontogeorgis, D. (2021). A proactive malicious software identification approach for digital forensic examiners. arXiv preprint.
https://arxiv.org/abs/2106.01863
6. Kara, I. (2022). Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges. Expert Systems with Applications, 214, 119133.
https://doi.org/10.1016/j.eswa.2022.119133
7. Lee, J.-H., & Kwon, H.-Y. (2022). Large-scale digital forensic investigation for Windows registry on Apache Spark. PLOS ONE, 17(12), e0267411.
https://doi.org/10.1371/journal.pone.0267411
8. Khalid, O., et al. (2023). An insight into the machine-learning-based fileless malware detection. Sensors, 23(2), 612.
https://doi.org/10.3390/s23020612
9. Zhang, S., et al. (2023). A malware detection approach based on deep learning and memory forensics. Symmetry, 15(3), 758.
https://doi.org/10.3390/sym15030758
10. Leng, T., et al. (2023). MemInspect: Memory forensics for investigating fileless attacks. In Proceedings of IEEE TrustCom (pp. 946–955).
https://doi.org/10.1109/TrustCom60117.2023.00134
11. Mutua, W. I. (2023). A model for advanced analysis of fileless malware threats through memory forensics (M.S. Thesis). United States International University–Africa.
https://erepo.usiu.ac.ke/handle/11732/6761
12. Dewan, R., & Venu, S. (2024). Detecting and investigating fileless malware: A deep dive. SSRN Electronic Journal.

https://doi.org/10.2139/ssrn.4932008
13. Sanjana, S., & Pavan, A. C. (2024). Unveiling the detection of fileless malware from dark web: A stealthy arsenal for web application exploitation. International Research Journal on Advanced Engineering and Management, 2(6).
https://doi.org/10.47392/IRJAEM.2024.0306
14. Singh, N., & Tripathy, S. (2025). Unveiling the veiled: An early stage detection of fileless malware. Computers & Security, 150, 104231.
https://doi.org/10.1016/j.cose.2024.104231
15. Sherazi, S. N. A., & Qureshi, A. (2025). Hybrid analysis model for detecting fileless malware. Electronics, 14(15), 3134.
https://doi.org/10.3390/electronics14153134
16. Arrieta, M. (2025). Hunting fileless malware in the Windows registry. Detect.fyi.
https://detect.fyi/hunting-fileless-malware-in-the-windows-registry/
17. Bentley, P. (2025). Analysis of Windows' registry key value to look for malware using AI generated code. University of Gloucestershire Discussion Paper.
https://eprints.glos.ac.uk/12960/
18. Zakaria, V., et al. (2025). Obfuscated file-less malware detection using integrating memory forensics data with machine learning techniques. Applied Computing and Informatics.
https://doi.org/10.1108/ACI-02-2025-0043
19. Joshi, S. B., & Warrier, R. R. (2025). Enhanced fileless malware detection using a deep learning approach. International Research Journal of Innovations in Engineering and Technology, 9(3), 221–227.
https://doi.org/10.47001/IRJIET/2025.903029
20. Yaseen, Q. M., Oudat, E., & Aldwairi, M. (2025). Efficient image-based memory forensics for fileless malware detection using texture descriptors and LIME-guided deep learning. Computers, 14(11), 467.
https://doi.org/10.3390/computers14110467
21. Rai, A., & Im, E. G. (2025). MemCatcher: An in-depth analysis approach to detect in-memory malware. Applied Sciences, 15(21), 11800.
https://doi.org/10.3390/app152111800
22. Zulkarnain, H., et al. (2025). Machine learning-driven detection of fileless malware in memory-resident attack vectors using volatile data analysis techniques.
https://doi.org/10.48550/arXiv.2503.XXXXX
23. MITRE ATT&CK®. (2025). Windows registry (Data source DS0037). MITRE Corporation.
https://attack.mitre.org/datasources/DS0037/
24. Joshi, M., Chincholkar, A., Londhe, A., Gole, P., & Mirgale, S. (2025). Enhancing cloud data security through AES encryption and SHA-256 integrity verification.
https://doi.org/10.48550/arXiv.2501.XXXXX
25. Kalavadekar, P. N., et al. (2025). Reinforcement learning to optimize tuberculosis screening strategies in resource-limited settings. Indian Journal of Tuberculosis.
https://doi.org/10.1016/j.ijtb.2025.01.003
26. Bornare, D., et al. (2025). Toward fair NLP models: Bias detection and mitigation in cloud-based text mining services.
https://doi.org/10.48550/arXiv.2502.XXXXX