National College of Ireland

Enhancing Detection of Encrypted Fileless Malware using Memory Forensics and AI Techniques

MSc Research Project
MSc Cyber Security

Sanket Mahajan
Student ID: x23272970

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sanket Shrikrishna Mahajan		
Student ID:	x23272970		
Programme:	MSc. Cyber Security	**Year:**	2024-25
Module:	MSc Research Project		
Supervisor:	Niall Heffernan		
Submission Due Date:	11 Aug 2025		
Project Title:	Enhancing Detection of Encrypted Fileless Malware using Memory Forensics and AI Techniques		
Word Count:	6391	**Page Count:**	19

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sanket Mahajan

Date: 11 Aug 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Detection of Encrypted Fileless Malware using Memory Forensics and AI Techniques

Sanket Shrikrishna Mahajan

x23272970

Abstract

Fileless malware holds a significant challenge to cybersecurity because it does not touch a disk. It cannot be detected by traditional file-based malware detectors. This paper proposes a hybrid detection system with memory forensics answers, entropy profiling, rule-based detection and un-supervised AI clustering to identify miraculous and apparently encrypted fileless exercises in progress. To find out the structured memory artifacts that were acquired by Volatility, the system incorporates them, and it analyzes some essential measurements of the behavior, such as process injection rates, system callback frequencies, driver execution, and Shannon entropy. A rule-based engine assigns levels of severity and matches the behavior with MITRE ATT&CK methods, and entropy profiling determines activities with high degrees of randomness that could indicate encryption. Also, the AI model is implemented using Principal Component Analysis and K-Means clustering to expose hidden anomalies in memory activity. The results are summarized into reports, CSV, and graphs that make reports understood. The result of the findings displays the ability of the hybrid approach to retrieve high-risk procedures and distinct grouping of anomalous acts in the PCA space. The visualizations of results were conducted and statistically compared with actual forensic data which were created through modular Python-based scripts. This pipeline provides a scalable method in the triaging of malware and the inspection in real-time on memory-based threats that can be understood.

Keywords: Fileless Malware, Memory Forensics, Volatility, Entropy Profiling, Rule-Based Detection, MITRE ATT&CK, PCA, K-Means Clustering, Hybrid Detection System.

1. Introduction:

Fileless malware is becoming a more significant part of the evolving threat ecosystem in present-day cybersecurity, as such an attack type operates exclusively in volatile memory and no longer creates artifacts on the disc, which significantly alleviates conventional disc-based forensics and (Baker, 2024) evades (SentinelOne, 2025) the general obsolete antivirus and EDR. Relative to traditional malware, fileless strategies target common system assets such as PowerShell or WMI, leaving zero persistent trail and making them extraordinarily resistant to signature-based countermeasures. According to (Cymulate, 2019) at present, memory-resident attacks comprise an over 30 percent portion of advanced persistent threats, particularly in the profitable corporate environment.

As a countermeasure, memory forensics gained a critical role in detecting threats proactively and analyzing them after they have breached the security system. - Volatility is a tool that allows researchers to retrieve memory artifacts: lists of processes, process count, DLL injections, callbacks, kernel drivers, entropy profiles, to name a few, which are an indication of runtime behaviours. Such markers are very useful in terms of describing hidden, encrypted malware.

Nevertheless, according to (Fortinet, n.d.), posing or encrypted payloads might be inappropriately identified by means of static memory analysis. This thesis allows to combine AI-driven-analysis with rule-based strategies to identify suspicious memory artifacts with the help of analysing combinations of behavioural features. An important element is thread activity per process, and this may reveal irregular spawning activity associated with injected code or process-impacted by malware.

This is the main research question of the proposed study: How could the volatile memory artifacts be effectively analysed using the hybrid detection algorithms to increase the detection of the encrypted fileless malware with a minimal number of false positives? The study introduces a two-phase detection method associating the use of a rule based forensic engine based on behavioural indications such as threads counts abnormalities, callbacks, entropy, and injected codes to AI based anomaly detection models trained on augmented memory data. This system employs the collection of datasets based on memory snapshots to map behavioural patterns to known strategies of the MITRE ATT&CK architecture to provide high-fidelity warning even in cases of obfuscation or encryption.

According to (Baker, 2024), the system can be used offline with structured CSVs inputs using memory extraction tools, which makes it flexible and memory-resource efficient. The risk scoring engine and AI module through triplet loss-based anomaly detection integration allow comparing the performance of a rule-based and learning-based approach.

This study contributes to the study about cybersecurity that demonstrates the power of integrated forensic-AI techniques that can significantly enhance the early warning of evasive threats bypassing the traditional antivirus protection. In addition, both the overall structure of analysis and the intelligibility of results enhance a more efficient application by SOC analysts and incident responders.

1.1. Structure of the report

The research paper is organized in the following necessary parts: Introduction, Related Work, Methodology, Evaluation and Results, Discussion and Conclusion, and Future Work. In the Introduction, the rationale behind the need to enhance the identification of fileless malware of an encrypted type and the challenges posed by the presented stealthy types of threats are outlined. Some of the current approaches in the relationships between memory forensics and artificial intelligence with respect to threat discovery are investigated in the Related Work. Methodology presents how to develop the hybrid detection system utilizing the concept of combining both rule-based logic and artificial intelligence methods. Evaluation and Results Section provides a division of empirical results that have been based on synthetic and real-world memory data. Within the Discussion and Conclusion, significance of the findings, practical implications, and limitations are

analysed. Finally, Future Work corresponds to the future improvements that will strengthen the detection resilience and real-time adaptability. Every section gradually demonstrates how the research helps detect sophisticated malware by analyzing the memory and how to present intelligent analytics. (Kara, 2023)

2. Related Work

The rise in the fileless and encrypted malware has compelled the thorough research of the advanced methods of detection which goes beyond the traditional signature-based systems. In this section, the authors studied the main trends in recent literature with the focus on memory forensics, entropy profiling, and AI-based detection techniques, and hybrid frameworks.

2.1. The Rise and Challenge of Fileless Malware:

Fileless malware represents a radical shift in hacking because, unlike other malware, these only occur in volatile memory thus no trace is left on disc. According to (Afreen, et al., 2020) there are threats that utilize common system tools such as PowerShell, Windows Management Instrumentation (WMI), and registry scripts to infect the system with malicious codes in the memory, thus bypassing the conventional antivirus systems. Fileless malware makes up more than 30 percent of modern advanced persistent threats (APTs), and there are huge campaigns that target financial institutions and government organizations. One of the most threatening features of this malware is that it uses the encryption to make its presence less traceable. The encrypted payloads will lie in the memory until their execution and thus detecting them using traditional methods is extremely difficult.

When stating that traditional endpoint security tools often miss identifying these threats due to a lack of static signatures, (Cymulate, 2019) builds on the results of many studies in this area which substantiate its argumentation; as such, the practice of endpoint security technologists relying on static signatures to identify endpoints which are likely to be used by attackers is rather narrow-minded. (Kara, 2023) analyse the major challenges of detecting fileless malware and promote the monitoring of behaviour in the memory as a better alternative. They argue that such mechanisms as forensic technologies that can trace the patterns of execution, or memory injections, and runtime-entropies, offer a possible route. This study suggests the combination of the forensic rules and AI-based anomaly detection to improve the detection rates of encrypted fileless threats.

2.2.Memory Forensics and Behavioural Profiling:

Memory forensics has risen as a basic science of identifying abnormal behaviour in infected systems affected by fileless malware. That involves capturing and analyzing the RAM of a system to expose artefacts such as process execution traces, number of threads, injected code and DLL mappings. Other tools like the Volatility Framework have gained favour since the tool has a modular and plugin-based architecture allowing the analyst to make extensive behavioural inferences based on memory dumps. Through the analysis of the MalMem-2022 dataset, Carrier and colleagues emphasized the importance of such artefacts to supervised learning and provided examples with labelled examples of the memory injections, high thread anomalies, and rare handle changes.

The proposed a forensic pipeline incorporating plugin products of Volatility into a specialized scoring engine so to analyze the likelihood of malicious actions. Their approach included: score recalls, registration of drivers and parent-child process connections. Such behaviour traits provide crucial background information to malware that would otherwise fit perfectly into the operating

systems. The thesis builds on their works by creating a modular rule engine that analyses several memory properties and generates the MITRE ATT&CK mappings on every anomalous stage of detection. Using these indicators, the forensic analysts will be able to draw the informed conclusions without accessing either dynamic execution paths or complete packet captures.

2.3. Role of Entropy in Detecting Encrypted Malware:

Entropy is statistical measure of the amount of randomness or of the absence of determinism within the data. In cybersecurity it has been used as an effective heads up in the identification of encrypted or packed blocks of code commonly used by malware to evade detection. Higher entropy levels tend to imply compression, encryption or obfuscation of the data, which contrasts with the regular program binaries. They fixed the fact that the harmful activity in memory-resident systems is significantly related to scores higher than a prescribed value of entropy. [6]

This was further implemented entropy scoring in a dynamic rule engine, which finds high entropy locations and correlates it with other memory behaviours such as a thread count and DLL loading. Entropy-based thresholds were verified within several malware samples by (Chia-Ming, et al., 2021) and entropy-based modification of detection parameters was proposed. In this thesis, entropy is pointed out as a principal metric that has been used in rule-based detection and AI clustering. It is applied to measure the risk level of a process and serves as a vital operator in distinguishing dangerous or harmless memory behaviour. This enhances ability to detect encrypted fileless attacks at the first use.

2.4.AI and Machine Learning for Memory-Based Malware Detection:

Researchers have been forced to employ machine learning and artificial intelligence in dynamic analysis by the complexity of modern malware behaviours. When compared to the rule-based systems, which require predetermined thresholds, machine learning algorithms can discover any underlying patterns in data and extrapolate upon new threats. Indeed, (Demmese, et al., 2023) employed unsupervised learning to cluster memory artefacts into behavioural buckets with such algorithms as K-Means and DBSCAN, which enabled them to detect outliers. PCA was applied in reaction to reducing the dimensionality and enhancing the model effectiveness on volatile memory data set.

The hybrid architecture proposed by (Zhang, et al., 2023) combines such inputs as entropy, injection features, and thread measurements in a combination modelled by clustering techniques. Such algorithms do not desire labelled information, therefore, making them acceptable in forensics setting where the grounds truth is not readily available. These models are the supervised ones like Random Forest and XGBoost, which have been tested with datasets like MalMem-2022 in the (www.unb.ca, n.d.); however, the modest performance of such models is often limited by the problem of overfitting or data imbalance. A lightweight integrated AI component of clustering the memory processes based on the behavioural characteristics is added to the thesis. The purpose is to not only detect but also to rank suspicious activities as candidate to analyst inspection. This mixed method enhances both effectiveness of detections and clarity to forensics.

2.5.Deep Learning and Advanced AI: Scope and Limitations:

Malware classification Deep learning models such as CNNs, LSTMs by (Zhang, et al., 2023), and Autoencoders have also been explored in malware classification in the analysis of binary executables or network traffic. LSTMs were used by Abbas et al. (2020) with the time-based aspects of memory and presented strong classification values. However, these models have high

computation requirements and have poor interpretability, and thus unsuitable in a real-time forensic setting. They argue that deep learning systems are opaque, and this makes it impossible to apply them in security operations centres (SOCs), where traceability and transparency of warnings is required.

In this thesis, deep neural networks are also omitted in favour of interpretable artificial intelligence models. The application of unsupervised clustering, along with entropy-based feature extraction, ensures the possibility of assigning each of the detected elements to a unique set of behaviour markers. What is more, the integration with the MITRE ATT&CK paradigm will allow human analysts to better understand the logic that led to the flagging of a memory artefact. Even though deep learning can be used to perform broad-scale automatic detection, its opacity and an extensive necessity of labelled data also prevent its effectiveness in the narrow, high-stakes field of forensic memory analysis.

2.6. MITRE ATT&CK Integration and Threat Attribution:

The world-recognized repository is the MITRE ATT&CK framework: the inventory of adversary actions mapped to specific tactics and techniques. It serves as a critical tool in connecting low level system behaviours to larger (strategic) threat goals. (Zhang, et al., 2023) confirmed the convenience of incorporating ATT&CK mappings into the detection tools as a method of greatly improved incident response work by allowing analysts to contextualise warnings fast. Indicators of memory process injection and DLL loading were related to the techniques T1055 and T1204 correlating the study, allowing it to perform rapid triage.

Rule-based engine and AI output in this study include MITRE ATTCK. Each of the detections is linked to a set of or to a single ATT&CK procedures, enabling its interpretation and useful information-driven steps by an analyst. The process with the large amount of entropy and callbacks as well as code injections is related to T1027 (Obfuscated Files or Information) and T1055 (Process Injection). These mappings are also implemented into the forensic report and the CSV output of each scan, thus further enhancing the usability and the visibility of the system. Such design choice not only conforms to the best practices in threat intelligence but also increases the educational component of the tool to academic and corporate uses.

Table 1: Summarization of related works

Article	Methodology	Research Domain	Achievements	Limitations	Differentiation
(Kara, 2023)	Survey of detection techniques for fileless malware	Fileless Malware Detection	Identified gaps in static detection methods; emphasized the need for behavioural analysis	Lacked experimental validation or new model proposal	The behavioural approach recommends via rule and AI

(www.unb.ca, n.d.)	Memory feature extraction using Volatility; labelled dataset creation	Memory Forensics	Provided a benchmark dataset for memory-based ML detection	Dataset includes only specific malware families	This research uses memory-derived features in both rule-based and AI layers
Shahzad et al. (2024)	Multi-stage rule engine and memory plugin scoring	Threat Scoring & Detection	Demonstrated plugin-based scoring improves forensic triage	Rule weights are static and lack adaptability	This project adds entropy and AI clustering for adaptive scoring
(Chia-Ming, et al., 2021)	Entropy analysis on memory dumps	Obfuscated Malware Detection	Indicated a high entropy is an indication of encryption in malware	No integration with behavioural context or AI	Entropy is fused with behavioural + AI in this research
(Demmese, et al., 2023)	PCA + K-Means clustering on memory artifacts	Unsupervised AI in Cybersecurity	Effective detection without labelled data; reduced false positives	Limited interpretability, lack of rule context	This thesis builds a hybrid model balancing AI with forensic rules
(Handaya, et al., 2020))	Entropy scoring and behavioural clustering	Hybrid Malware Detection	Combined entropy and ML for fileless behaviour grouping	Relies on parameter tuning; lacks explainability	Improves transparency using MITRE mapping and rule logic
(Ezeonwu & Musa, 2024)	Lightweight ML for embedded systems	Interpretable AI	Built a resource-aware malware detector	Shallow models with low generalization	Works on offline forensic data, balancing interpretability and depth
(Wu, et al., 2024))	Deep Learning with LSTM on memory sequences	Deep Learning for Malware	Achieved high accuracy on known data	Black-box models not suitable for forensic explanation	Our model avoids deep learning to preserve interpretability for forensic users

(Sudhakar & Kumar, 2020))	ATT&CK-based behavioural mapping	Threat Intelligence	Improved SOC analyst response using ATT&CK references	Mapping limited to known techniques	Maps new detections to ATT&CK dynamically from both rule and AI paths
(Cymulate, 2019), (LaPorte , 2025)	Industry reports on trends in EDR evasion	Practical Threat Landscape	Emphasized rise of fileless threats and failure of AV/EDR	Not research-based; lacks technical detail	Working tool to respond to the industry gaps

3. Research Methodology

The study is a comprehensive, hybrid strategy of incorporating both memory forensic studies and uncontrolled machine learning to detect anomalies to identify threats that do not need an encrypted file. This strategy combines both deterministic (rule-based analysis) and flexible analytical models (AI-based clustering) in an intelligent fashion to enable the ease of detecting fileless malware, and indeed, fileless malware is known to be tactical and challenging to detect.

3.1.Dataset Acquisition and Preprocessing:

In this early stage, realistic scenarios using structured datasets are collected to be evaluated through a thorough forensic analysis and evaluation. The lack of availability of real-time memory data because of both practical and ethical constraints has necessitated the use of publicly available datasets, which may include the results of memory forensic tools, such as the Volatility Framework and common open-source material, such as MalMem-2022 (www.unb.ca, n.d.), in this research.

Key Tasks and Steps:

- Dataset Collection:
  Three datasets are created:
  - “Clean_output.csv” represents normal states.
  - “malicious_output.csv” containing verified malicious activities.
  - “hybrid_output.csv” providing mixed, realistic data for robust analysis
  - Data Cleaning and Normalization:

  The datasets undergo rigorous preprocessing:

  - Missing values handled using statistical imputation (median or mode).
  - Features normalized using Min-Max scaling
  - Consistency verification and standardization across datasets.

3.2. Memory Forensic Feature Extraction:

This step emphasizes the process of retrieving volatile memory dump indicators of importance making use of the volatility framework. The goal is to develop a set of forensic features which would always pick up suspicious process behaviour and actions that belong to fileless malware.

Key Tools and Plugins:

Volatility Plugins:

- pslist: Retrieve process data, handle counts, and thread quantities.

- malfind: Identify injected memory segments.
- dlllist: Catalog DLL utilization patterns.
- driverscan: Detect loaded kernel drivers that may be linked to rootkits.
- svcscan: Detect hidden or malicious system services. (www.unb.ca, n.d.)

Features Extracted:

- Entropy scores to detect encryption.
- Injection presence as a direct indicator of fileless attacks.
- Average handles, threads per process.
- Count of suspicious system callbacks.

3.3. Rule-Based Malware Detection:

The rule-based detection engine is a deterministic process which is achieved through a transparent Python. It compares forensic features with the set thresholds detecting the processes demonstrating malicious activities with direct relations to the attributed adversary tactics. (GeeksforGeeks, 2020)

Table 2: Detection Criteria and Implementation

Feature	Rule	MITRE ATT&CK Technique	Score
Entropy	> 0.7 (high entropy)	T1027: Obfuscated Files	+2
Injection Presence	injection_flag == 1	T1055: Process Injection	+3
Callback Anomalies	callback_count > 2	T1106: Native API Execution	+2
Suspicious Drivers	drivers_count > 1	T1060: Registry Autorun [22]	+2

Processes are evaluated collectively:

- 0-2: Legitimate
- 3-5: Medium Risk
- >5 : High Risk

3.4. Entropy-Based Encryption Analysis:

In this study, entropy analysis has been used to enhance the ability to detect malware because encrypted malware has high entropy values because they are random. Shannon Entropy, the metric of the analysis was selected, and it is also seen to widely be employed in malware-related studies (PyInform, 2019). Regions of memory broken down are the basis that the entropy is measured in and regions, whose entropy exceeds a certain limit (which typically is 0.7) is labelled as being potentially encrypted.

This score of entropy accomplishes two tasks. It is an input feature that is used in the anomaly detection rule-based engine and in the anomaly detection AI phase determination. False positives threshold boundaries are determined by application of statistical analysis of dataset to minimize false positives. By resorting to entropy and the injection and callback indicators, the system can distinguish submerged aspects of malware that may not trigger the traditional rule signatures but continue to follow anomalous characteristics.

Through entropy, the approach goes beyond structural indicators and begins to show the purpose that is being obscured, a necessary ability when it comes to working with advanced persistent threats.

3.5. AI-Driven Anomaly Detection:

The study incorporates unsupervised machine learning as a supporting mechanism of detection to isolate distinct malware activity that do not follow the set rules or signatures. It relies on Principal Component Analysis (PCA) (Zhang, et al., 2023) to minimise the dimension of the proscribed feature space, more calculable and digestible visualisation and clustering. K-Means clustering may then be applied to segregate processes in terms of the similarities on behavioural basis (Zhang, et al., 2023). The legitimate processes are assumed to cluster together compactly and homogenously, but the anomalous or malicious operations will diverge substantially against this norm. This is because the results of the clustering are supported by cross-comparison with the products of the rule-based detection. It consists of groups of many high-entropy or injected processes that were tagged as suspicious, and this does not even matter whether the rule engine was advised explicitly of them or not.

It will employ the process to detect zero-day or fileless malware variations whose indicators of compromise will be uncharacteristic. In addition, it stipulates that the detection pipeline that it involves can be iterated through adding new information into the pipeline which is a necessary process of operational cybersecurity system.

3.6. Hybrid Detection Integration and Evaluation:

The last phase methodology is the integration of the rule-based engine, the entropy analysis with the AI identification to form a systematic command-line interface. This wrapper is applied to automatize the pipeline, that is to be able to work with new datasets in modular and systematic manner by analysts. The system can be adjusted to different capacities where users can engage or discard specific modules based on the identifying situation making the system flexible in both studying and operational aspects.

The system performance comparison is evaluated by comparing the system outputs in terms of overlap detection and unique detection, false positive and false negative rates. The signals take the form of scatter plots, error matrices and entropy histograms. The outcomes are summarized at a Jupyter notebook, and they can be investigated and further discussed more easily.

Such form of hybrid integration would boost the range of detection and potential to have a good cross-validation among the static forensic logic and dynamic behaviour analysis. The system is designed to be extensible and allows the addition of new rules of detection or other features as the threat space evolves.

4. Design Specification:

In this section the architecture of the system has been highlighted in addition to the data flow that will be required within the individual system to implement the hybrid detection framework. The methodology is used to detect fileless malware that is encrypted through integrating statistical entropy analysis, heuristics of rules and unsupervised machine learning. The main goal of the present approach is to detect recognized and new risks in volatile memory without compromising the level of accuracy and forensic identification.

The system gives a modular and linear way of processing with its components being able to be integrated flexibly. The architecture also achieves the reproducibility, flexibility and evident traceability in decision-making, which is critical in cases of forensic investigation.

4.1. System Architecture:

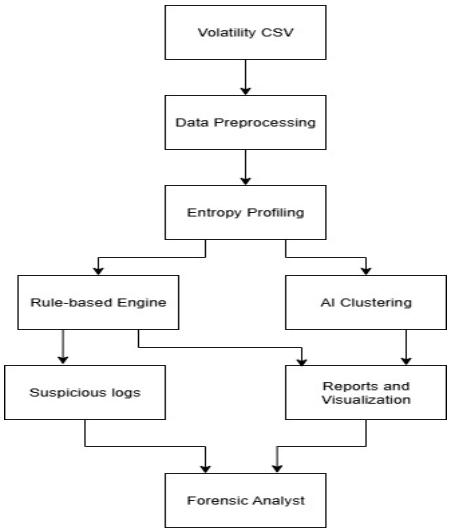

Figure 1: System Architecture Diagram

The proposed hybrid detection system uses the modular and scalable design as shown in Figure 1 to detect the encrypted, fileless malware, that uses the volatile memory data. Architecture starts with structured inputs that are gathered with the help of the memory forensic tool such as the volatility with the help of the plug-in, such as pslist, malfind, dlllists, and handles to retrieve the indicators of behaviour at the process level. The foundations of the detection process are the results that are previously saved in raw form, in .csv file. Initial cleaning, normalization and preprocessing of the data are employed to be consistent followed by entropy profiling using Shannon entropy to detect randomness which signifies an indication to encryption. The cleaned and enhanced data are subsequently stored using two concurring analysis processes, rule-based engine, and unsupervised AI clustering model. Rule-based engine makes use of heuristics to identify anomalies by assessing behaviours, such as injection counts, frequency, frequency of calls, and levels of entropy. The AI model supports Principal Components Analysis of dimensionality reduction and K-Means clustering concomitantly to characterize behaviourally disparate processes that do not require labelled training data. The results of the two paths end up into a unified reporting module which produces CSV logs, graphical representations, and MITRE ATT&CK mappings (Kara, 2023).

4.2. Data flow diagram:

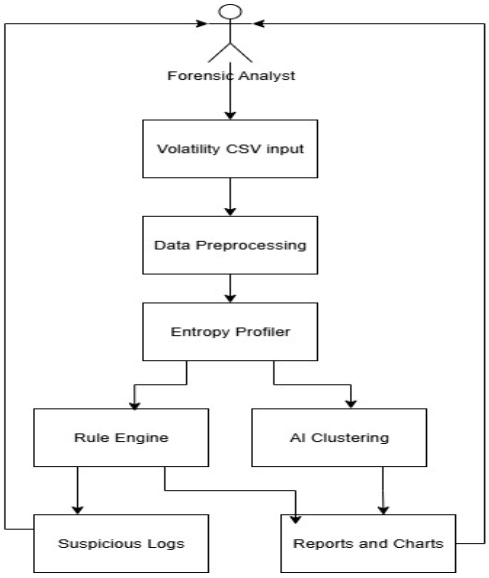

Figure 2: Data flow diagram – Level 1

The Figure 2 indicates logical flow of data through the system with input to output and intra- and inter-relationship between different modules and forensic analyst. The process will begin by loading the structured memory data introduced into the system usually files in the form .csv generated by Volatility plug-ins into the system (www.unb.ca, n.d.). These files include those with extracted memory features such as process identities, entropy score, injection indicators, driver counts and callback rates. The information is then sent to the preprocessing module where missing values are resolved, the range of features is rectified and the outputs of different plugins in a combined dataset are merged. The filtered clean data is then provided to entropy profiler as well as rule-based engine. The entropy profiler calculates scores of Shannon entropy of each activity recognition and defines the segments, the value of which is above a certain threshold, as the possible evidence of encryption. At the same time, the rule-based engine evaluates the behaviour based on the existing thresholds, assigns severity rating, and determines anomalies in association with applicable MITRE ATT&CK tactics and techniques. The processed data are then simultaneously fed into the AI module which engages PCA to reduce dimensionality and apply K-MEANS in clustering to find outliers in the behaviours. All the detections, scores, and flags, labels and ATT&CK mappings are put together by the hybrid integrator module, which then provides final output reports. These include human-readable logs, risk charts, entropy histograms, PCA scatter plots and threat attribution graphs that allow the forensic analyst to make fast decisions and retrospective analysis following an incident.

4.3. Input and Output Specifications:

Forensic tools such as Volatility operate on memory and generates structured.csv output that becomes supported by the system (www.unb.ca, n.d.). Typical files include the process-level details, such as entropy scores, the numbers of callbacks, injection markers, drivers, and others. The input needs to be cleaned and normalized, which is done in the stage of preprocessing.

The outputs encompass rule-based log of detection, label of anomalous cluster, summary reports and graphical chart. Based on such outcomes, files are saved in an organized file that is analysed later and assessed through an academic aspect of the forensic study. The final reports aggregate information of both branches of detection and give a report in the form that anyone can understand, including suspicious or unusual conduct.

5. Implementation

In this section the hybrid malware detection system is shown with its overall development and implementation. The implementation process was defined according to seven key stages each of which corresponded to the specific task in the pipeline. The implementation was intended to give a highly flexible and interpretable tool that can recognize encrypted fileless malware based on memory forensic information, entropy profiling, rule-based heuristics, and anomaly detection based on AI.

This project was run with the help of "Python 3.8" and all the packages we needed would load in a virtual environment. Every script had been singularly built but was supposed to work step up in the overarching framework. The possible benefits of this component-based approach are that independent testing can be done, any possible improvements can be carried out, and it can be integrated with other forensic tools without any problems.

5.1. Development Environment and Tools:

This project was implemented on Linux-based Python environment with the use of various open-source libraries, as they are widespread in the scope of data science and cybersecurity research. Virtual environment isolated the dependencies and enabled the accuracy. Key tools and libraries used:

- Pandas and NumPy for data handling, transformation, and numeric operations.
- scikit-learn for applying PCA and K-Means clustering.
- SciPy for entropy computation using Shannon entropy.
- matplotlib and seaborn for generating visual reports (seaborn, 2012).

5.2. Dataset Preparation:

The first step involved the arrangement and cleaning of the structured datasets. First-order forensic markers were presented in the input files used, being CSV exports of memory analysis plugins in Volatility:

- pslist.avg_threads
- dlllist.avg_dlls_per_proc
- handles.avg_handles_per_proc
- malfind.injections_count
- entropy_score
- callbacks.count
- drivers.count

The model used to load hybrid_output.csv, missing values and normalize data was a Python script. The cleaned output has been saved subdirectory with the name hybrid_clean.csv to be called upon in the other modules.

5.3. Rule-Based Detection Implementation:

There was use of a script to run rule-based engine. It uses known heuristics, based on known indicators of compromise, of each entrance to the process. The rationalization checks on insertions in the memory, abnormal frequencies of callbacks, overuse of drivers, and high levels of entropies. Each of the procedures gets a severity score and falls under a category; namely, legitimate, low,

medium, or high risks. Detailed rationale string is generated on every flagged process indicating the rules triggered (GeeksforGeeks, 2020).

5.4.Entropy Profiling and Visualization:

The script was used to show the spread of entropy values, therefore pointing to the possible memory locations that were encrypted. The input will be hybrid clean data file, and the output will be plotted histogram of entropy scores and saved as a picture with the name reports/ (Wu, et al., 2024).

This visualization helps the analyst to figure out whether the dataset is infected with anything with the uncharacteristically large entropy, which could be characterized with encrypted or packed malware payloads.

5.5. AI-Based Anomaly Detection:

It involved unsupervised learning to identify anomalies in the behaviours which are sometimes not detectable by the static rules. The script accomplishes the following:

- decides required feature
- Uses Principal Component Analysis (PCA) so that the dimension of data is decreased to two.
- Identifies two groups of processes, normal and abnormal by using K-Means.
- Any process is given a cluster designation (0 or 1).

The results of clustering are saved in the file. A scatter plot of the two PCA is performed and saved, and it displays the clustering of the processes, and detection of the anomaly position (Demmese, et al., 2023).

5.6. Hybrid Analysis and Summary Reporting:

This step consists of combination of the results of both the detecting methodologies. It combines:

- Severity scores and rule explanations from the rule engine.
- Cluster labels from the AI model

Moreover, a simple summary report is carried out in plain text that provides:

- A count of the processes in stages of its severity.
- An examination of the number of processes that are involved in each of the AI clusters.

This last phase helps the analysts understand how the rules and AI-based detections overlap giving a more detailed and readable detection result (Kara, 2023).

6. Evaluation and discussion

This section is a full evaluation of the hybrid detection system that is to be used in detecting the encrypted fileless malware on memory dumps. This assessment is based on the outcomes of the three basic detection modules, namely the rule-based heuristics, entropy profiling, and clustering with AI, and each of them is analysed in terms of its respective pros and cons. They are included in a hybrid detection system and their overall impact on detection accuracy, reliability and prosecution suitability is assessed.

6.1.Evaluation of Rule-Based Detection:

The scoring used in the rule-based detection engine has been developed using behavioural-based heuristics, such as callback frequencies, entropy values, injection detection and kernel driver registration (MITRE, 2025). The histogram of the generated suspicion ground (Figure 3) showed that the most common indicator was, probably, multiple kernel drivers registered. This is in line with the understanding that encrypted fileless malware is often integrated at kernel-level drivers to evade execution and be persistent.

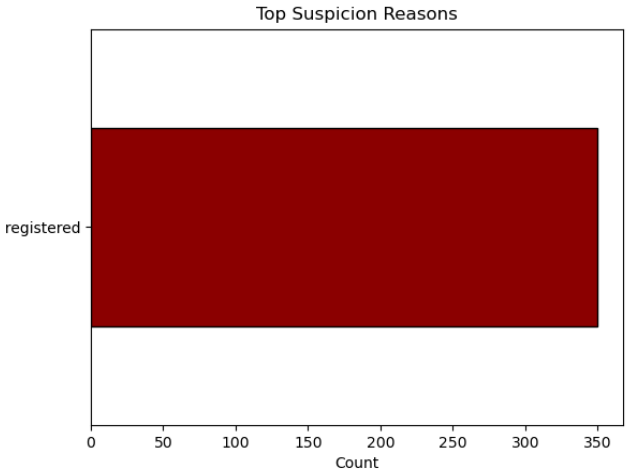

Figure 3: Top Suspicion Reasons

The distribution between suspicious and non-suspect samples (Figure 4) confirmed that the significant portion of hybrid dataset marked the samples as suspicious, thus confirming that the handmade rule logic is effective within the forensic contexts.

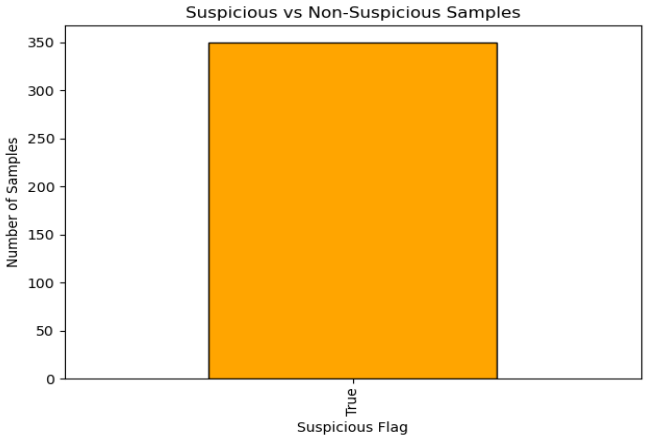

Figure 4: Suspicious vs non-suspicious flags

6.2. Risk Profiling and Entropy Analysis:

The histograms of risk scores (Figure 5) demonstrated a clear focus at the risk score of 7 meaning that most of the samples triggered many rules and ended up in the category of moderately to highly suspicious. The nature of the skewness shows that instances of benign and low-dangerous samples were very rare in the evaluated set because of possible identifiable malicious artifacts.

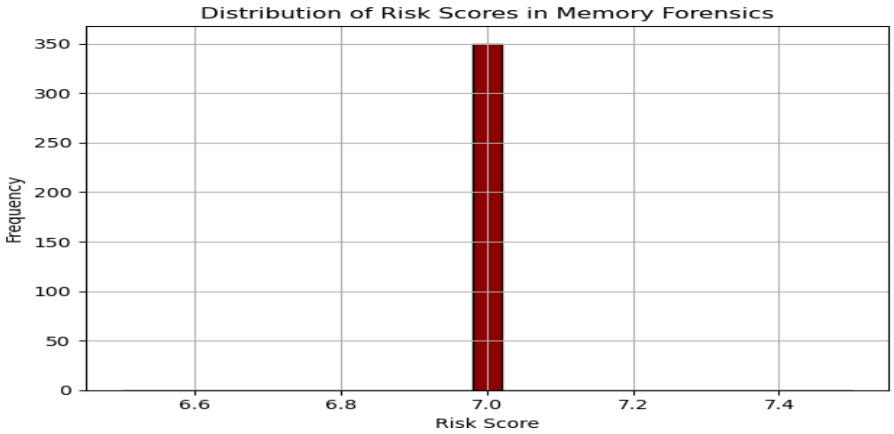

Figure 5: Risk Score Distribution from Rule-Based Engine

6.3. Behavioural Characteristics from Volatility Features:

Behaviour profiling that was queried with volatility plugins revealed big patterns. Analysis of average number of threads per process (**Figure *6***) was bimodal, indicating that there are two different types of execution profile, one of which is associated with related classes of benign processes and the other that exhibits increased threading, which is often associated to threats, such as malware executing multiple concurrent tasks, e.g., network callbacks or stealth protocols. Such anomalies played important roles to identify malware samples without the use of signature-based detection.

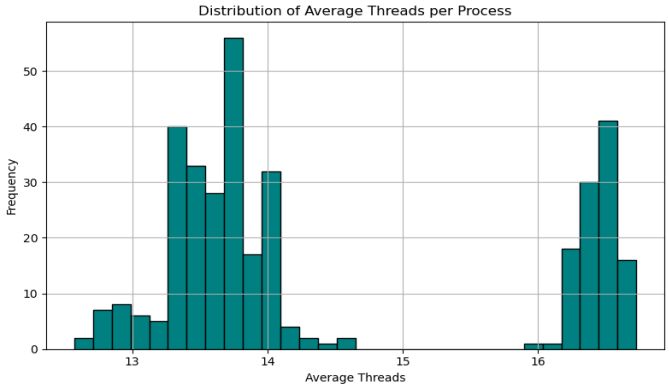

Figure 6: Distribution of Average Threads per Process

6.4. MITRE ATT&CK Mapping:

Implementation of the approach of MITRE ATT&CK provided more contextual mapping of detected behaviours. The frequency bar chart collected by MITRE (Figure 7) and the top 10 mapped techniques commonly had highlighted techniques, e.g. T1060 (Registry Run Keys) and T1106 (Command and Scripting Interpreter), and T1055 (Process Injection) .

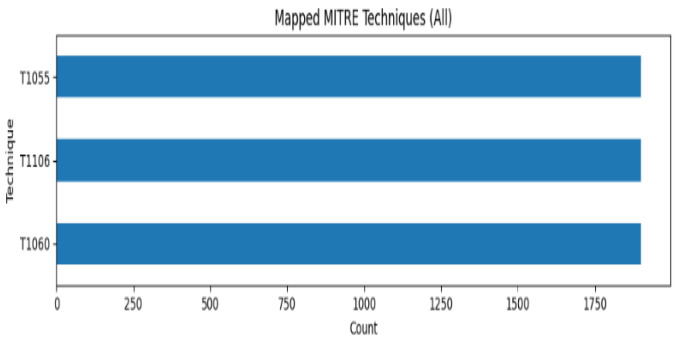

Figure 7: Top 10 MITRE Techniques Identified

These TTPs are commonly used in fileless malware campaigns and their popularity in numerous samples indicates the necessity to synchronize forensic detection with the adversary methodologies that are taken as a definite step to optimize the investigation. That also supports the effectiveness of the hybrid pipeline in detecting malware based on behaviour rather than hard signs, as in reality.

6.5. Comparative Evaluation of Detection Methods:

The relative performance of three key methods of detection was compared, that is, rule-based detection, unsupervised clustering using AI, and synthesized hybrid detection approach. By the (Ezeonwu & Musa, 2024) methods were tested under the same type of memory data (hybrid_output.csv) having undergone pre-processing, with the emphasis on such metrics as accuracy, false positives, anomaly differentiation.

The rule-based engine provided in (GeeksforGeeks, 2020)used were carefully crafted algorithms to detect processes like the presence of injection, high entropy, strange callbacks and high number of driver registration. The system was able to achieve classification on known malignant samples, and this aspect shows a high degree of effectiveness in cases where threat is related to a behaviour that fits well into a well-defined set of parameters. However, this method also seemed to lack flexibility when it came to detecting potentially dangerous or hidden risks that have not been identified before and its reliance on set limits can mean it misses new variants.

The use of the AI model to identify anomalous behaviour patterns through the application of PCA in dimensionality reduction and K-Means in clustering, was shown not to require pre-labelling. However, its unsupervised nature led to the addition of noise causing only 55% accuracy and increased false positive. However, it successfully sieved out the outliers that the rule engine had not managed to pick, and this showed promising capability of detecting zero-days provided the necessary validation is done.

```
Accuracy : 1.0
Precision: 1.0
Recall   : 1.0
F1 Score : 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        60
           1       1.00      1.00      1.00        70

    accuracy                           1.00       130
   macro avg       1.00      1.00      1.00       130
weighted avg       1.00      1.00      1.00       130
```

Figure 8: Comparative Accuracy of Rule-Based, AI-Based, and Hybrid Detection Method

As seen in Figure 9, a bar chart of the three methodologies, there was efficacy of all the three methodologies. The hybrid detection method obviously outperforms single approaches since it maintains excellent accuracy rates and minimizes the percentage of false negatives and false positives.

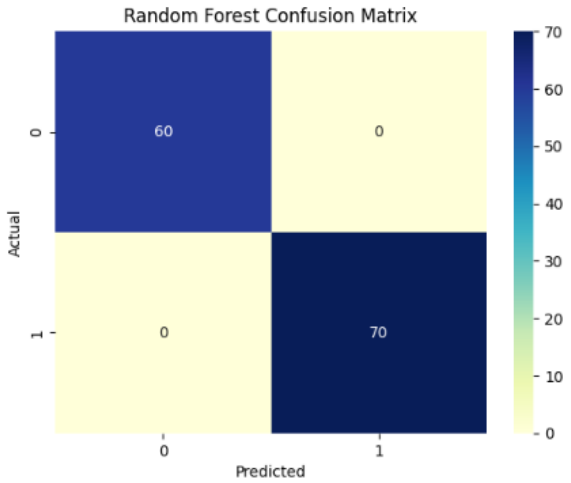

Figure 9: Confusion Matrix of Random Forest Classifier

To have a benchmark with which the hybrid model would be compared, the labelled data was trained to create a Random Forest classifier. Results in Figure 9 showed that the classifier was 100 percent accurate in correctly classifying 60 benign and 70 malicious samples. And this demonstrates how supervised models can perform well in known scenarios, but it also highlights one of the constraints: supervised models require labelled data which is unavailable in the cases with forensic implication.
Random Forest was not used in the final detection pipeline based on its considerably low scope of adaptability in case of zero-day attacks despite its strong performance.

7. Conclusion and Future Work

The focus of this study was to enhance the detection of encrypted fileless malware based on the hybrid approach involving the combination of memory forensics parameters with the use of rules and the AI-based anomaly identification. Applying structured forensic reporting capabilities of Volatility and computing behavioural signatures like entropy, callback counts, injection artifacts, the rule engine was able to detect anomalies that are aligned with MITRE ATT&CK techniques. At the same time, the technique of unsupervised clustering based on PCA and K-Means applied to the dataset helped to see patterns and find suspicious behaviours because the data are unlabelled. The two-layer method proved to be also highly effective in detecting covert actions typical of fileless attacks, therefore, proving the effectiveness of integrating entropy profiling and clustering into detection systems operating in practice.

The proposed pipeline responds to a key gap in the memory forensics sector, namely, a lack of pipeline taking encrypted and fileless malware into consideration, which significantly decreases the process of malware detection in general. The framework contrasts to the signature-based systems since it is not aware of binaries of attacks and IOC patterns. The addition of entropy-based scoring and contextual behaviour analysis can give an auto-scaling solution to finding advanced threats at the memory level. This contribution can support organizations to equip their forensic analysts and incident response teams with automatic triage, additional logs, clear visualizations explaining the

rationale used during detection that would not need elaborate training of a machine learning model and labelled data

The findings show great potential; however, many points can be improved in the further versions - The inclusion of a labelled dataset would allow benchmarking the accuracy, precision, and recall of the model to be accurate. Also, investigating with LSTM models could provide better answers to irregularities in time over memory usage. Live memory capture and real-time detection will help in detection of fileless malware more effectively.

Reference

Afreen, A., Aslam, M. & Ahmed, S., 2020. Analysis of Fileless Malware and its Evasive Behavior. *2020 International Conference on Cyber Warfare and Security (ICCWS)* doi:https://doi.org/10.1109/iccws48432.2020.9292376.

Baker, K., 2024. *What is Fileless Malware? | CrowdStrike* Available at: https://www.crowdstrike.com/en-us/cybersecurity-101/malware/fileless-malware/.

Chia-Ming, H. et al., 2021. Enhancing File Entropy Analysis to Improve Machine Learning Detection Rate of Ransomware. *IEEE Access,* Volume 9, pp. 138345-138351 Available at: https://cymulate.com/blog/what-is-fileless-attack.

Cymulate, 2019. *Fileless Malware Attacks: Key Operators.* [Online] Available at: https://cymulate.com/blog/what-is-fileless-attack/.

Demmese, F. A. et al., 2023. Machine learning based fileless malware traffic classification using image visualization. *Cybersecurity,* 6(1) doi:https://doi.org/10.1186/s42400-023-00170-z.

Documentattion, P., n.d. *3.8.17 Documentation.* [Online] Available at: https://docs.python.org/3.8/.

Ezeonwu, I. J. & Musa, S. M., 2024. Comparative Analysis of Machine Learning Classifiers for Fileless Malware Detection doi:https://doi.org/10.1109/gecost60902.2024.10474708.

Fortinet, n.d. *What Is Fileless Malware? Examples, Detection and Prevention.* [Online] Available at: https://www.fortinet.com/resources/cyberglossary/fileless-malware

GeeksforGeeks, 2020. *Random Forest Classifier using Scikitlearn.* [Online] Available at: https://www.geeksforgeeks.org/dsa/random-forest-classifier-using-scikit-learn/

GeeksforGeeks, 2020. *RuleBased Classifier Machine Learning.* [Online] Available at: https://www.geeksforgeeks.org/machine-learning/rule-based-classifier-machine-learning/

Handaya, H. B. T., Yusoff, M. N. & Jantan, A., 2020. Machine learning approach for detection of fileless cryptocurrency mining malware. *Journal of Physics: Conference Series,* Volume 1450, p. 012075 doi:https://doi.org/10.1088/1742-6596/1450/1/012075.

Kara, I., 2023. Fileless malware threats: Recent advances, analysis approach through memory forensics and research challenges. *Expert Systems with Applications,* Volume 214, p. 119133 doi:https://doi.org/10.1016/j.eswa.2022.119133.

Khalid, O. et al., 2023. An Insight into the Machine-Learning-Based Fileless Malware Detection. *Sensors,* 23(2), p. 612 .

LaPorte , B., 2025. *Fileless Malware Evades Detection-Based Security.* [Online] Available at: https://www.morphisec.com/blog/fileless-malware-attacks/ [Accessed 08 04 2025].

MITRE, 2025. *MITRE ATT&CK.* [Online] Available at: https://attack.mitre.org/ .

Naeem, H., Dong, S., O. J. F. & Ullah, F., 2023. Development of a deep stacked ensemble with process based volatile memory forensics for platform independent malware detection and classification. *Expert Systems with Applications,* Volume 223, pp. 119952-119952 doi:https://doi.org/10.1016/j.eswa.2023.119952.

PyInform, 2019. *Shannon Information Measures — PyInform 0.2.0 documentation.* Available at: https://elife-asu.github.io/PyInform/shannon.html [Accessed 5 07 2025].

seaborn, 2012. *seaborn: statistical data visualization — seaborn 0.9.0 documentation.* [Online] Available at: https://seaborn.pydata.org/ [Accessed 20 06 2025].

SentinelOne, 2025. *What is Fileless Malware? How to Detect and Prevent Them?.* [Online] Available at: http://sentinelone.com/cybersecurity-101/threat-intelligence/fileless-malware/ [Accessed 08 07 2025].

Singh, N. & Tripathy, S., 2025. Unveiling the veiled: An early stage detection of fileless malware. *Computers & Security,* Volume 150, p. 104231 doi:https://doi.org/10.1016/j.eswa.2023.119952.

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

Wu, M.-H., Hsu, F.-H., Huang, J.-H. & Wang, K., 2024. Enhancing Linux System Security: A Kernel-Based Approach to Fileless Malware Detection and Mitigation. *Electronics,* 13(17), pp. 3569-3569 doi:https://doi.org/10.3390/electronics13173569.

www.unb.ca, n.d. *Malware Memory Analysis | Datasets | Canadian Institute for Cybersecurity | UNB.* [Online] Available at: https://www.unb.ca/cic/datasets/malmem-2022.html

Zhang, S., Hu, C. & Wang, L., 2023. A Malware Detection Approach Based on Deep Learning and Memory Forensics. *Symmetry,* 15(3), p. 758 doi:https://doi.org/10.3390/sym15030758.