

CHAPTER 1 - INTRODUCTION

1.1 Background

In recent years, the cyber threat landscape has shifted from predominantly **file-based malware** toward **fileless** and **Living-off-the-Land (LOTL)** tactics that minimize disk artifacts and instead abuse built-in Windows capabilities. These intrusions frequently rely on legitimate components such as **PowerShell**, **Windows Management Instrumentation (WMI)**, **MSHTA**, or **signed Windows utilities** to execute payloads in memory and blend into normal administrative activity, making them harder to detect using conventional signature- and rule-based defenses (CrowdStrike, 2024).

This trend is particularly impactful in modern enterprises where Windows remains widely deployed and where administrators routinely use powerful native tooling. Adversaries exploit that same tooling as “**LOLBin-style**” execution mechanisms to proxy malicious actions ranging from initial execution and persistence to lateral movement and command-and-control while appearing operationally legitimate. MITRE ATT&CK documents these behaviours in techniques such as **PowerShell abuse (T1059.001)** and **System Binary Proxy Execution (T1218)**, highlighting how trusted binaries and scripts can be repurposed to evade traditional defenses (MITRE ATT&CK, 2025).

Recent industry reporting further indicates that adversaries increasingly succeed without relying on traditional malware binaries, emphasizing the operational reality that many intrusions now “look” like ordinary system activity at the log level. For example, CrowdStrike reports a high proportion of detections involving **malware-free** tradecraft (i.e., actions carried out through legitimate tools rather than dropped executables), reinforcing the need for detection approaches that focus on **behavioral context** and **command semantics** rather than file signatures alone (CrowdStrike, 2024).

Consequently, recent academic work (2023-2025) increasingly emphasizes **log- and telemetry-driven detection**, including hybrid approaches that combine behavioural/context

features (process lineage, timing, parent–child patterns) with models that capture **command semantics** to improve robustness against obfuscation and “living-off-the-land” execution patterns. This motivates detection designs that treat attacks as **behavioural sequences and relationships** observed through telemetry rather than static files especially under real-time monitoring constraints (Khalid et al., 2023).

1.2 Research Problem

Although several advanced malware detection techniques had been proposed, most existing frameworks remained limited by static, context-unaware, and offline designs. Conventional endpoint detection systems typically analyzed logs or artifacts only after execution, resulting in delayed responses to malicious activities. These weaknesses were particularly critical in the case of fileless and LOTL attacks, which executed directly in memory and leveraged legitimate system binaries such as PowerShell and WMI to evade signature-based detection (Sudhakar and Kumar, 2020).

First, current detection models generally failed to capture contextual dependencies such as parent–child process relationships, privilege escalation chains, and cross-log event correlations. Two identical PowerShell commands could represent very different intentions depending on which process invoked them, under which user account, and as part of which process lineage. Many existing machine learning-based methods treated each log event independently and therefore ignored the graphical structure of process interactions (Filar and French, 2020).

Second, most prior frameworks lacked semantic understanding of command intent. They might flag suspicious keywords or encoded content, but they did not fully interpret the combination of tokens, arguments, and execution patterns that distinguished benign automation from malicious LOTL activities. This limitation reduced their ability to robustly detect obfuscated or encoded scripts.

Third, explainability mechanisms were rarely integrated into real-time detection pipelines. Where explainable AI (XAI) techniques such as SHAP or attention visualization had been explored, they were often applied in post-hoc analysis rather than during live detection. As a result, analysts received alerts without clear insight into the underlying reasoning, making it difficult to validate, prioritize, or fine-tune detections (Manthena et al., 2025).

Finally, many existing approaches depended heavily on a single telemetry source (e.g., Sysmon

only or ETW only), which made them vulnerable to evasion through log tampering, configuration weaknesses, or visibility gaps. There was a lack of unified frameworks that combined multiple complementary telemetry streams in a coherent, real-time architecture.

Based on these observations, the core research problem addressed in this thesis was:

The absence of a real-time, explainable detection framework that combines contextual and semantic understanding from multi-source Windows telemetry to effectively identify and interpret fileless and LOTL attack behaviours.



Table A.1

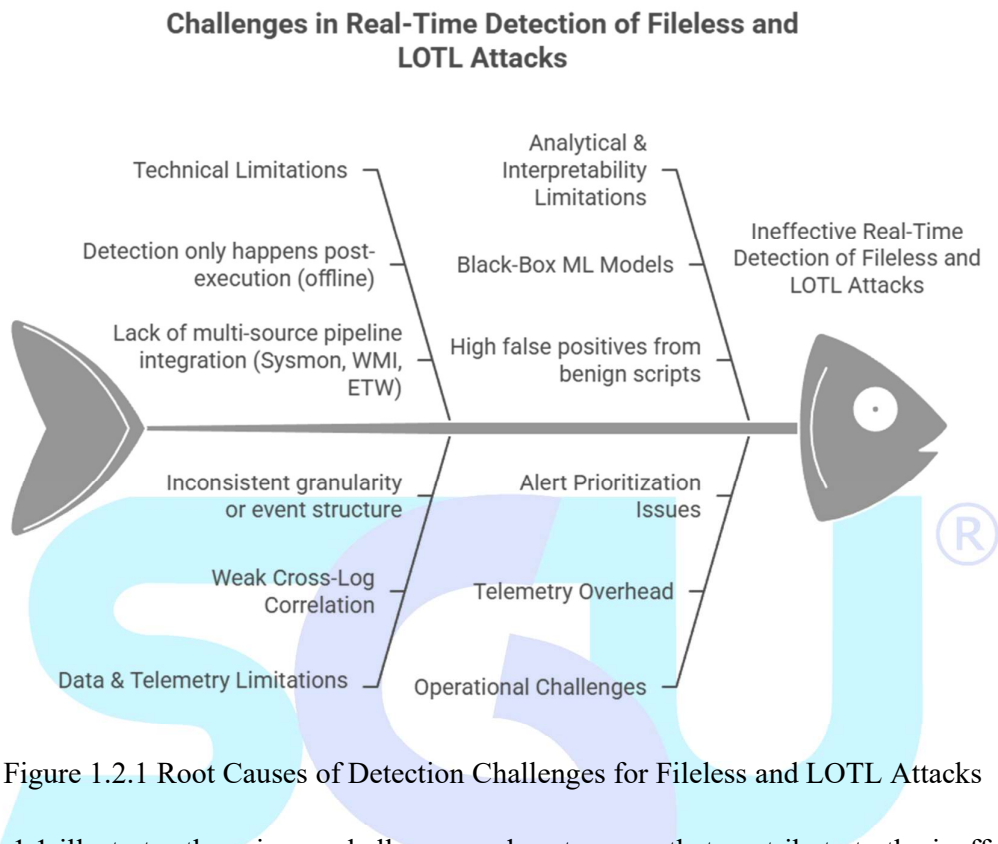


Figure 1.2.1 Root Causes of Detection Challenges for Fileless and LOTL Attacks

Figure 1.1 illustrates the primary challenges and root causes that contribute to the ineffective real-time detection of fileless and Living-off-the-Land (LOTL) attacks. The diagram highlights three main categories of limitations identified from previous studies: technical, data and telemetry, and analytical interpretability issues. Technical challenges arise from the dominance of offline and post-execution detection frameworks, as well as the absence of multi-source pipeline integration across Sysmon, WMI, and ETW telemetry. Data-related issues concern weak event correlation, inconsistent log granularity, and the lack of privilege or lineage context necessary for accurate process-level understanding. Analytical limitations involve the black-box nature of existing machine learning models, which lack explainability and frequently misclassify benign administrative scripts as malicious behavior. In addition, several operational challenges such as telemetry overhead and alert prioritization issues further degrade real-time performance. Collectively, these gaps justify the development of a context-semantic real-time detection framework, as proposed in this research.

1.3 Research Questions

To address the research problem described above, this study was guided by the following research questions:

1. **RQ1.** How could contextual and semantic information be effectively fused from multi-source Windows telemetry (Sysmon, WMI, and ETW) to enable real-time detection of fileless and LOTL attacks?
2. **RQ2.** What kind of hybrid architecture could accurately model both process context and command semantics to distinguish between benign and malicious LOTL behaviours in real time?
3. **RQ3.** How could explainable artificial intelligence (XAI) techniques, such as attention mechanisms and SHAP, be integrated to enhance interpretability and analyst trust in the detection process?
4. **RQ4.** To what extent did the proposed context-semantic detection framework improve detection accuracy, latency, and explainability compared to existing state-of-the-art LOTL detection approaches?

1.4 Research Objectives

In line with the research questions, this thesis pursued the following objectives:

1. **To design** a multi-source telemetry fusion pipeline that integrated contextual and semantic information from Sysmon, WMI, and ETW in real time, forming a unified data foundation for behavior analysis.
 2. **To develop** a hybrid analytical architecture that combined contextual reasoning and semantic analysis to classify benign and malicious LOTL activities based on process-level relationships and command-line behavior.
 3. **To incorporate** Explainable Artificial Intelligence (XAI) techniques including attention-based visualization and SHAP (SHapley Additive Explanations) in order to provide interpretable insights into the model's detection decisions.
 4. **To evaluate** the proposed framework's performance in terms of detection accuracy, false positive rate, latency, and interpretability, and to compare its effectiveness with existing state-of-the-art LOTL detection approaches.
-

1.5 Research Hypotheses

Based on the research questions and objectives, the following hypotheses were formulated:

1. **H1:** The integration of multi-source Windows telemetry (Sysmon, WMI, and ETW) significantly improved the accuracy of detecting fileless and LOTL attacks compared to single-source detection methods.
2. **H2:** A hybrid analytical model that combined contextual and semantic representations outperformed conventional machine learning models in identifying fileless and LOTL attacks.
3. **H3:** The integration of Explainable Artificial Intelligence (XAI) mechanisms, specifically attention visualization and SHAP, improved interpretability and analyst confidence in the model's detection outcomes.
4. **H4:** The proposed context-semantic detection framework achieved higher overall performance, measured in terms of detection accuracy, latency, and interpretability, than current state-of-the-art fileless and LOTL detection frameworks.

1.6 Scope and Limitation

This research focuses on the development and evaluation of a **Context-Semantic Real-Time Detection Framework** designed to identify fileless and Living-off-the-Land (LOTL) attacks in Windows environments. The study specifically investigates how the fusion of contextual and semantic features from Windows telemetry can enhance detection accuracy, latency, and interpretability.

1.5.1 Scope of the Research

- The study focused on **Microsoft Windows** environments, where fileless and LOTL attacks predominantly exploited native components such as PowerShell, WMI, and registry mechanisms.
2. The framework utilized **three primary telemetry sources**: Sysmon, WMI, and ETW, collected via a unified shipper agent for real-time analysis.
 3. The detection methodology concentrated on **fileless and LOTL behaviors** and did not explicitly target traditional file-based malware such as classic ransomware or trojans.
-

4. Performance evaluation included metrics such as **detection accuracy, false positive rate, latency, and interpretability**.

4.5.1 Limitations of the Research

1. The proposed framework was **limited to Windows platforms** and was not directly generalized to Linux or macOS without telemetry adaptation.
2. Continuous multi-source telemetry collection introduced **computational and storage overhead**, which could affect scalability in very large deployments.
3. The experimental dataset combined controlled LOTL scenarios and selected public datasets; therefore, it might not fully represent the diversity of fileless techniques observed in large-scale enterprise environments.
4. While XAI components enhanced interpretability, they also incurred **additional computational cost**, potentially impacting throughput when explanations were generated for every event in real time.

1.7 Thesis Structure

This thesis is structured into five chapters, each addressing a specific aspect of the research process and collectively supporting the development of the proposed context-semantic real-time detection framework for fileless and Living-off-the-Land (LOTL) attacks in Windows environments.

- **Chapter 1: Introduction**

Presented the research background, problem formulation, research questions, objectives, hypotheses, scope, and limitations, and outlined the overall structure of the thesis.

- **Chapter 2: Literature Review**

Reviewed theoretical foundations and recent studies related to fileless and LOTL attacks, log-based and telemetry-based detection methods, machine learning and semantic modeling, and explainable AI in cybersecurity.

- **Chapter 3: Research Methodology**

Described the research design, multi-source data acquisition from Sysmon, WMI, and ETW, preprocessing and feature extraction pipelines, the proposed context-semantic framework architecture, and the evaluation procedures used to validate the framework.

- **Chapter 4: Results and Analysis**

Presented the implementation details, experimental setup, and empirical results, including detection performance, latency, and explainability outcomes. The chapter discussed how the findings addressed the research questions and hypotheses.

- **Chapter 5: Conclusion and Recommendation**

Summarized the main findings and contributions of the research, outlined its limitations, and proposed directions for further work, such as cross-platform extensions, scalability improvements, and integration with enterprise security operations.

