

CHAPTER 2 - LITERATURE REVIEW

2.1 Business and Industry landscape

2.1.1 The Rise of Digital Interdependence

Modern enterprise environments are increasingly characterized by deep digital interdependence, where business operations rely on tightly coupled information systems, cloud services, endpoint devices, and automated workflows. This transformation has enabled organizations to achieve unprecedented levels of operational efficiency, scalability, and responsiveness. However, prior studies indicate that such interdependence also introduces complex security dependencies that are difficult to monitor and control using traditional protection models (Khalid et al., 2023).

Enterprise systems today rarely operate in isolation. Endpoints are continuously integrated with centralized identity services, remote management infrastructures, and automated administrative frameworks. Academic research on fileless and Living-off-the-Land (LOTL) attacks demonstrates that attackers increasingly exploit these legitimate operational mechanisms to execute malicious actions without introducing new binaries or persistent files into the system (Harish and Swapna, 2024; Meher et al., 2025). As a result, benign administrative activity and malicious behaviour often share overlapping execution paths, making distinction based solely on static indicators unreliable.

Several studies emphasize that the widespread use of scripting engines and system management tools such as PowerShell, Windows Management Instrumentation (WMI), and command-line interpreters has fundamentally altered the threat landscape (Harish and Swapna, 2024; Khalid et al., 2023). Rather than relying on exploit-driven payload deployment, adversaries can leverage built-in trust relationships and native execution environments to operate stealthily within enterprise systems. This behaviour-centric attack strategy significantly expands the effective attack surface beyond traditional network perimeters.

From a detection perspective, digital interdependence introduces systemic risk. A single compromised user context or abused administrative interface can provide attackers with lateral reach across interconnected systems. Empirical studies on fileless malware consistently report that malicious activity often manifests as subtle deviations in execution behaviour rather than explicit anomalies, requiring detection approaches that reason over context, process relationships, and temporal patterns (Elongha et al., 2025; Pal et al., 2025).

Industry threat intelligence reports corroborate these findings by highlighting the increasing prevalence of stealth-driven attacks that exploit legitimate system tools and trusted execution paths. However, such commercial observations primarily reinforce conclusions already established in academic literature and are therefore best treated as contextual background rather than primary evidence (CrowdStrike, 2024).

In summary, the rise of digital interdependence has blurred the boundary between legitimate operational behaviour and malicious activity. This shift challenges traditional artifact-centric detection models and motivates the need for behaviour-aware, context-sensitive, and semantically informed detection frameworks capable of operating in real-time enterprise environments.

2.1.2 Threat Actors Targeting Modern Enterprise Systems

Modern enterprise systems have become primary targets for a wide range of threat actors due to their high operational value, extensive digital interconnectivity, and reliance on trusted administrative mechanisms. Academic studies consistently report that contemporary cyber adversaries increasingly prioritize stealth, persistence, and operational longevity over rapid exploitation, particularly in enterprise environments where detection latency can be exploited to achieve strategic objectives (Khalid et al., 2023).

Advanced Persistent Threat (APT) groups represent a significant class of actors targeting modern enterprises. Unlike opportunistic attackers, APT groups operate with long-term objectives such as espionage, intellectual property theft, and strategic disruption. Empirical research demonstrates that these actors frequently leverage fileless and Living-off-the-Land (LOTL) techniques to evade traditional security controls by abusing legitimate system utilities rather than deploying standalone malware binaries (Harish and Swapna, 2024; Meher et al., 2025).

Beyond nation-state linked APT groups, financially motivated cybercriminals and ransomware operators have also adopted LOTL methodologies. Studies on enterprise intrusion campaigns reveal that administrative tools such as PowerShell, Windows Management Instrumentation (WMI), and scheduled task utilities are commonly misused to facilitate lateral movement, credential access, and command execution while maintaining a low forensic footprint (Elongha et al., 2025). This convergence of techniques across threat actor categories further complicates attribution and detection.

A key challenge in defending against these actors lies in the overlap between benign administrative activity and malicious behaviour. Academic analyses highlight that the same tools used by system administrators for maintenance and automation are frequently exploited by attackers, resulting in execution patterns that appear legitimate at the surface level (Harish and Swapna, 2024; Khalid et al., 2023). Consequently, detection strategies that rely solely on known indicators of compromise or static signatures are insufficient against such adversaries.

Industry threat intelligence reports corroborate academic findings by documenting a steady increase in enterprise-targeted attacks that exploit trusted system components and native execution environments. However, these reports primarily reinforce trends identified in peer-reviewed research and are therefore best interpreted as contextual evidence rather than primary analytical foundations (CrowdStrike, 2024; Microsoft, 2023).

In summary, modern enterprise systems are targeted by diverse threat actors that increasingly employ fileless and LOTL techniques to blend malicious actions with legitimate system behavior. This evolution underscores the need for detection approaches that emphasize behavioral context, process relationships, and semantic understanding of system activity rather than artifact-centric analysis (Khalid et al., 2023; Meher et al., 2025)

2.1.3 Operational Pressure and Security Complexity

Modern enterprises operate under increasing operational pressure driven by demands for high availability, rapid service delivery, and continuous system automation. To meet these requirements, organizations rely heavily on administrative scripting, remote management tools, and automated workflows that reduce manual intervention and improve efficiency. Academic studies indicate that while such practices enhance productivity, they also introduce substantial

security complexity by expanding the number of privileged execution paths within enterprise environments (Khalid et al., 2023).

As enterprise infrastructures scale, security teams are required to monitor an ever-growing volume of system events generated by endpoints, applications, and management frameworks. Empirical research highlights that the sheer scale and velocity of operational telemetry often exceed the capacity of traditional security mechanisms to perform fine-grained inspection in real time (Elongha et al., 2025). Consequently, many detection systems prioritize efficiency over contextual reasoning, potentially allowing subtle malicious activities to blend into normal operational noise.

The complexity is further amplified by the dual-use nature of administrative tools. Tools such as PowerShell, Windows Management Instrumentation (WMI), and task schedulers are essential for legitimate system maintenance, yet they are equally capable of being abused by attackers to execute malicious commands, establish persistence, or perform lateral movement. Multiple studies report that the frequent and legitimate invocation of these tools makes anomaly detection particularly challenging, as malicious executions often exhibit minimal deviation from normal administrative behavior (Elongha et al., 2025; Khalid et al., 2023).

Operational constraints also impose strict requirements on detection latency. In enterprise environments, delayed detection can result in significant impact, including credential compromise, lateral propagation, and prolonged attacker dwell time. Research on fileless and LOTL attacks emphasizes that effective defense mechanisms must operate under real-time or near-real-time constraints while maintaining low false positive rates to avoid disrupting critical business operations (Elongha et al., 2025; Harish and Swapna, 2024)

Industry analyses echo these challenges by documenting the growing burden placed on security operations centers (SOCs), where analysts must triage large volumes of alerts under time pressure. However, such commercial observations primarily contextualize operational realities already identified in academic research and should therefore be interpreted as supplementary evidence rather than primary analytical sources (CrowdStrike, 2024; Microsoft, 2023).

In summary, operational pressure and security complexity form a reinforcing cycle in modern enterprise systems. Increased automation and reliance on trusted administrative tools expand the attack surface while simultaneously constraining detection capabilities. These conditions

necessitate security approaches that can reason over behavioral context and execution semantics in real time without imposing prohibitive computational or operational overhead (Elongha et al., 2025; Khalid et al., 2023).

2.1.4 Emergence of Stealth-Driven Attacks in Enterprises

In recent years, enterprise cyber threats have increasingly shifted toward stealth-driven attack strategies that prioritize evasion, persistence, and low observability over rapid system compromise. Unlike traditional malware campaigns that rely on overt exploit delivery and file-based payloads, modern attackers aim to remain undetected for extended periods by blending malicious actions with legitimate system behavior. Academic studies consistently report that this shift is particularly evident in enterprise environments, where complex operational workflows provide fertile ground for concealment (Khalid et al., 2023).

Stealth-driven attacks frequently leverage fileless execution techniques and Living-off-the-Land (LOTL) binaries to minimize their forensic footprint. By executing malicious logic directly in memory and abusing trusted system utilities, attackers significantly reduce the presence of static artifacts that can be detected through signature-based or hash-based mechanisms. Empirical research demonstrates that such attacks often evade conventional endpoint security controls, allowing adversaries to establish persistence and conduct lateral movement with minimal detection (Harish and Swapna, 2024; Meher et al., 2025).

A defining characteristic of stealth-driven enterprise attacks is the strategic misuse of native administrative tools. Tools such as PowerShell, Windows Management Instrumentation (WMI), and system schedulers are routinely exploited to execute commands, download payloads, or manipulate system configurations under the guise of legitimate administrative activity. Multiple studies emphasize that the semantic content of these executions may appear benign in isolation, while their behavioral context such as execution timing, process lineage, and interaction patterns reveals malicious intent (Khalid et al., 2023; Leng et al., 2024).

The effectiveness of stealth-driven attacks is further amplified by obfuscation and polymorphism techniques. Attackers commonly employ command encoding, script fragmentation, and dynamic payload generation to evade static inspection and rule-based detection. Research on fileless malware indicates that these techniques substantially degrade the effectiveness of traditional detection systems, necessitating approaches that can reason over

both behavioral patterns and semantic representations of execution activity (Leng et al., 2024; Meher et al., 2025).

Industry reports corroborate the academic consensus by documenting a steady rise in dwell time and attacker persistence within enterprise networks. However, these observations primarily serve to contextualize the operational impact of stealth-driven attacks and should be interpreted as complementary to peer-reviewed research rather than as primary analytical evidence (Google, 2023).

In summary, the emergence of stealth-driven attacks represents a fundamental evolution in the enterprise threat landscape. By exploiting legitimate system tools, minimizing detectable artifacts, and leveraging obfuscation, attackers challenge conventional detection paradigms. These developments underscore the need for detection frameworks that integrate behavioral context, process-level reasoning, and semantic analysis to effectively identify malicious activity in real-time enterprise environments (Harish and Swapna, 2024; Khalid et al., 2023).

2.1.5 Implication for Malware and Threat Detection Research

The evolution of enterprise cyber threats toward stealth-driven, fileless, and Living-off-the-Land (LOTL) techniques has significant implications for malware and threat detection research. Traditional detection paradigms, which primarily rely on static signatures, file artifacts, and known indicators of compromise, are increasingly inadequate in environments where malicious activity closely resembles legitimate operational behaviour (Khalid et al., 2023).

Academic research consistently highlights that modern attacks often manifest as subtle behavioral deviations rather than explicit anomalies. As a result, detection research must move beyond isolated event analysis toward approaches capable of reasoning over execution context, process relationships, and temporal patterns. Studies on fileless malware emphasize that understanding *how* system activities are executed is often more informative than analyzing *what* artifacts are present on disk (Elongha et al., 2025; Khalid et al., 2023).

Another critical implication is the growing importance of semantic interpretation in detection systems. As attackers abuse scripting languages and command-line interfaces, the semantic structure of commands and scripts becomes a valuable source of intelligence. However, multiple studies note that semantic analysis alone is insufficient when detached from execution

context, as benign administrative scripts may share syntactic and lexical similarities with malicious ones. This observation suggests that effective detection requires the integration of semantic understanding with contextual behavioural analysis.

Operational constraints further shape the direction of detection research. Enterprise environments demand real-time or near-real-time detection capabilities that can operate under high event throughput while maintaining low false positive rates. Research in this area indicates that computationally intensive models or approaches requiring extensive offline analysis may be impractical for deployment in production systems, despite their theoretical effectiveness (Harish and Swapna, 2024).

Collectively, these findings imply that future malware and threat detection research should prioritize hybrid approaches that combine multiple complementary perspectives, including behavioral context, semantic reasoning, and efficient decision mechanisms. Such approaches are better positioned to address the challenges posed by stealth-driven enterprise attacks while remaining practical for real-world deployment (Leng et al., 2024).

2.2 Business Challenges: Physical vs Cyber Threats

2.2.1 Characteristics of Physical Threats

Physical threats refer to risks that originate from tangible, observable actions against physical assets, infrastructure, or personnel. Examples include theft, vandalism, sabotage, unauthorized physical access, and natural hazards. Such threats are typically constrained by geographical boundaries and require a direct physical presence or interaction with the targeted environment. Prior research in risk management literature emphasizes that physical threats are generally easier to localize, observe, and contain compared to cyber threats, due to their inherent dependence on physical proximity and visibility (ISO/IEC, 2022).

A defining characteristic of physical threats is their comparatively linear and deterministic nature. Physical attacks often follow a predictable sequence of actions, such as reconnaissance, access, and execution, which can be mitigated through established controls including surveillance systems, access restrictions, and physical security policies. Studies on organizational risk assessment highlight that physical threat detection relies heavily on human observation, environmental sensors, and clearly defined perimeter defenses (NIST, 2012).

Another important aspect of physical threats is the clarity of attribution and impact. In many cases, the source of a physical attack can be identified through direct evidence, such as surveillance footage or eyewitness accounts. Additionally, the effects of physical incidents such as damage to equipment or facilities are immediately observable and localized. This direct cause-and-effect relationship simplifies incident response and recovery processes when compared to cyber incidents, which may unfold over extended periods.

From a risk management perspective, physical threats are typically addressed through preventive and reactive measures that are well understood and mature. Frameworks for physical security emphasize deterrence, detection, and response mechanisms that operate within clearly defined temporal and spatial constraints. While physical threats remain relevant to enterprise operations, multiple studies argue that their impact and scalability are inherently limited when compared to digitally mediated threats (Longo et al., 2025).

Industry analyses support this distinction by noting that while physical security incidents continue to occur, their potential for large-scale, simultaneous impact across distributed systems is significantly lower than that of cyber-attacks. Such observations provide contextual background for understanding why modern enterprises increasingly prioritize cyber threat detection over traditional physical risk mitigation (ISO/IEC, 2022).

2.2.2 Characteristics of Cyber Threats

Cyber threats differ fundamentally from physical threats in terms of scale, visibility, and operational dynamics. Unlike physical attacks, cyber threats can be executed remotely, often without requiring direct interaction with the targeted environment. Academic research consistently emphasizes that this lack of physical constraints allows cyber-attacks to scale rapidly across geographically distributed systems, enabling a single attacker or campaign to impact multiple organizations simultaneously (ISO/IEC, 2022).

A defining characteristic of cyber threats is their high degree of stealth and ambiguity. Cyber attacks frequently exploit legitimate system functions, trusted software components, and authorized user privileges to conceal malicious intent. Studies on enterprise intrusions indicate that malicious cyber activity often unfolds gradually, blending into normal operational behaviour and remaining undetected for extended periods (Khalid et al., 2023). This stealth-

oriented nature complicates detection and attribution, as attackers deliberately minimize observable indicators of compromise.

Another key aspect of cyber threats is the complexity of attribution and impact assessment. Unlike physical incidents, where cause and effect are typically immediate and observable, cyber-attacks may involve multi-stage execution chains, lateral movement, and delayed payload activation. Research highlights that attackers increasingly leverage fileless techniques and Living-off-the-Land (LOTL) tools to avoid leaving persistent artifacts, further obscuring forensic analysis and incident reconstruction (Leng et al., 2024)

Cyber threats are also characterized by their adaptability and rapid evolution. Attack techniques can be modified, obfuscated, or repurposed with minimal cost, allowing adversaries to respond quickly to defensive measures. Empirical studies show that attackers frequently reuse legitimate administrative tools and scripting environments, modifying command structures and execution sequences to evade detection while maintaining functional effectiveness.

Industry analyses reinforce these observations by reporting a growing prevalence of stealthy, low-noise attacks that exploit trusted system components rather than overt malware payloads. However, as with physical threat comparisons, such industry reports primarily contextualize trends already established in academic literature and should be treated as supplementary background (Microsoft, 2023).

In summary, cyber threats are distinguished by their remote execution, scalability, stealth, and adaptive nature. These characteristics fundamentally challenge traditional detection paradigms and underscore the need for security approaches that emphasize behavioural context, process-level reasoning, and semantic interpretation rather than reliance on static indicators alone.

2.2.3 The Cyber Risk Convergence Problem

The cyber risk convergence problem refers to the increasing overlap between physical, cyber, and operational risks within modern enterprise environments. Traditionally, organizations managed physical security and cyber security as separate domains, each governed by distinct risk models, controls, and response mechanisms. However, academic research indicates that digital transformation has blurred these boundaries, creating interconnected risk surfaces where

cyber incidents can directly trigger physical, operational, and financial consequences (FSB, 2022)

In converged enterprise environments, cyber threats no longer remain confined to information systems alone. Compromised credentials abused administrative tools, and unauthorized remote access can lead to cascading impacts across business operations, industrial processes, and safety-critical systems. Studies on enterprise risk management highlight that cyber attacks increasingly exploit trust relationships between digital systems and operational workflows, thereby bypassing traditional perimeter-based and domain-specific defenses.

A key challenge arising from cyber risk convergence is the inadequacy of legacy threat modelling approaches. Physical risk models typically assume localized impact, clear attribution, and observable events, while cyber risk models often focus on network-centric or asset-centric indicators. Research demonstrates that modern attacks particularly fileless and Living-off-the-Land (LOTL) techniques operate across these conceptual boundaries, combining digital stealth with operational legitimacy to evade detection (Leng et al., 2024).

The convergence problem also complicates detection and response strategies. Security monitoring systems designed around isolated domains may fail to correlate subtle signals that span process behaviour, user activity, and system interactions. Empirical studies emphasize that attackers increasingly leverage this fragmentation by distributing malicious actions across multiple benign-looking events, each insufficient to trigger alerts in isolation (Khalid et al., 2023).

Industry analyses reflect similar concerns by reporting that security operations teams struggle to maintain situational awareness across converged risk domains. Nevertheless, such reports primarily contextualize challenges already identified in academic literature and should be interpreted as supporting observations rather than foundational evidence (e.g., Gartner, 2023; ENISA, 2023).

In summary, the cyber risk convergence problem underscores a fundamental shift in the enterprise threat landscape. As physical, cyber, and operational risks become increasingly intertwined, detection research must move toward holistic approaches capable of correlating behavioural, contextual, and semantic signals across traditionally separate domains.

2.2.4 Why Cyber Threats Now Outweigh Physical Risks

In modern enterprise environments, cyber threats increasingly outweigh physical risks in terms of scale, impact, and persistence. While physical threats are constrained by geographical proximity and require direct interaction with targeted assets, cyber threats can be executed remotely, repeatedly, and at scale. Academic research consistently emphasizes that this asymmetry enables attackers to inflict disproportionate damage with relatively low cost and minimal exposure, making cyber-attacks a more dominant risk factor for contemporary organizations.

Another critical factor is the persistence and dwell time associated with cyber threats. Unlike physical incidents, which are often immediately detectable, cyber intrusions may remain undetected for extended periods while adversaries conduct reconnaissance, escalate privileges, and move laterally within enterprise systems. Studies on fileless and Living-off-the-Land (LOTL) attacks demonstrate that attackers deliberately exploit this detection latency to maximize operational impact, often leveraging legitimate system tools to avoid raising suspicion (Khalid et al., 2023).

Cyber threats also surpass physical risks in their potential for systemic and cascading effects. A single compromised endpoint or user account can serve as an entry point for widespread disruption across interconnected systems, cloud services, and business processes. Research on enterprise cyber incidents highlights that such cascading effects are rarely observed in physical attacks, where damage is typically localized and easier to contain (FSB, 2022).

Furthermore, the adaptability of cyber threats contributes to their increasing dominance. Attackers can rapidly modify techniques, obfuscate execution patterns, and repurpose legitimate tools in response to defensive measures. Empirical studies indicate that this continuous evolution enables adversaries to bypass static defenses and traditional detection models, reinforcing the imbalance between cyber and physical risk management capabilities.

Industry analyses reinforce these conclusions by reporting a sustained increase in cyber incident frequency and impact relative to physical security breaches. However, as with earlier sections, such reports primarily serve to contextualize trends already established in peer-reviewed research and should be interpreted as supplementary background rather than primary analytical evidence (Verizon, 2025)

In summary, cyber threats now outweigh physical risks due to their scalability, persistence, systemic impact, and adaptive nature. These characteristics fundamentally challenge traditional security assumptions and necessitate a shift in research focus toward detection approaches that can address stealthy, context-dependent, and rapidly evolving cyber-attacks within enterprise environments.

2.2.5 Implications for Malware Research

The increasing dominance of cyber threats over physical risks has profound implications for malware research. As attackers exploit scalability, stealth, and legitimate system functionalities, traditional assumptions underlying malware detection such as the presence of identifiable binaries or overt exploit behaviour are no longer sufficient. Academic studies consistently demonstrate that modern malware campaigns, particularly fileless and Living-off-the-Land (LOTL) attacks, challenge artifact-centric detection models by operating within trusted execution environments.

One major implication is the need to reconceptualize malware as a behavioural phenomenon rather than a static object. Research indicates that malicious activity increasingly manifests through sequences of actions, process interactions, and contextual deviations that may appear benign when examined in isolation. This shift necessitates malware research approaches that emphasize behavioural modelling, contextual correlation, and temporal reasoning instead of relying solely on static indicators or isolated events (Khalid et al., 2023).

Another important implication concerns the role of semantic understanding in malware detection. As attackers abuse scripting languages and command-line interfaces, the meaning and structure of executed commands become critical sources of insight. However, multiple studies caution that semantic analysis alone cannot reliably distinguish malicious intent from legitimate administrative activity without complementary behavioural context. This observation highlights the need for research that integrates semantic interpretation with execution context and process-level relationships.

Practical deployment considerations further shape the direction of malware research. Enterprise environments impose strict constraints on detection latency, computational overhead, and false positive tolerance. Research in this area emphasizes that detection approaches must balance analytical sophistication with operational feasibility, particularly for real-time monitoring

systems operating at scale (Harish and Swapna, 2024). Consequently, malware research must prioritize methods that are both effective and deployable under realistic enterprise conditions.

In summary, the evolving threat landscape requires malware research to move beyond traditional, artifact-focused paradigms toward integrated detection strategies that combine behavioral, contextual, and semantic perspectives. These implications form the foundation for subsequent discussions on malware characteristics, evolution, and detection techniques presented in the following sections.

2.3 Malware Overview

Malware has long been recognized as one of the most pervasive and impactful forms of cyber threats, serving as the foundation of many intrusion campaigns observed in enterprise environments today. As digital infrastructures evolve, malware has also undergone significant transformation shifting from simple, predictable payloads toward sophisticated, stealth-oriented attack chains capable of evading modern detection systems. To understand why fileless and Living-off-the-Land (LOTL) techniques have gained prominence, it is essential to examine malware from a historical, conceptual, and operational perspective.

2.3.1 Defining Malware in the Modern Context

Malware has traditionally been defined as malicious software designed to disrupt, damage, or gain unauthorized access to computer systems. Early definitions of malware largely focused on file-based artifacts such as viruses, worms, and trojans that could be identified through static signatures, known hashes, or recognizable binary patterns. While this definition remains valid in a historical sense, academic research increasingly argues that it is insufficient to capture the full spectrum of modern malicious activity observed in enterprise environments (Khalid et al., 2023).

In contemporary threat landscapes, malware is no longer confined to standalone executable files. Instead, malicious behaviour often emerges through the abuse of legitimate system components, trusted administrative tools, and in-memory execution techniques. Studies on fileless malware emphasize that modern attacks may not introduce any persistent artifacts to disk, yet still achieve full malicious objectives such as command execution, persistence, and

lateral movement (Singh and Tripathy, 2025). This evolution fundamentally challenges artifact-centric definitions of malware.

Another defining characteristic of modern malware is its behavioural and contextual nature. Rather than being identifiable by static properties alone, malicious activity is increasingly characterized by *how* system resources are used, *when* processes are executed, and *how* different components interact over time. Research highlights that many malicious executions closely resemble benign administrative actions at the surface level, with intent only becoming apparent when examined through execution context and process relationships (Leng et al., 2024).

The growing reliance on scripting languages and command-line interfaces further complicates malware definition. PowerShell scripts, batch commands, and other interpreted code can serve both legitimate administrative purposes and malicious objectives. Semantic analysis of such scripts provides valuable insights into intent; however, multiple studies caution that semantic features alone are insufficient without contextual behavioural information, as benign and malicious scripts may share similar syntactic structures (Elongha et al., 2025).

Consequently, modern malware is increasingly understood not as a static entity, but as a dynamic sequence of actions that exploit legitimate system functionality to achieve malicious goals. This perspective motivates malware research to adopt definitions that emphasize behaviour, context, and semantics rather than exclusive reliance on file-based artifacts or isolated execution events.

2.3.2 Evolution of Malware Across Four Eras

The evolution of malware reflects broader changes in computing environments, defensive technologies, and attacker objectives. Rather than remaining static, malware has continuously adapted to evade detection mechanisms and exploit emerging system capabilities. This evolution can be broadly categorized into four overlapping eras: file-based malware, obfuscated and packed malware, fileless malware, and Living-off-the-Land (LOTL) attacks.

The earliest era of malware was dominated by **file-based malicious executables**, where attackers distributed standalone binaries designed to infect systems, replicate, or cause direct disruption. Detection during this period primarily relied on signature-based antivirus techniques

that matched known byte patterns within malicious files. While effective against known threats, this approach struggled to keep pace with the growing diversity of malware variants (Khalid et al., 2023).

As defensive capabilities improved, attackers transitioned to the second era characterized by **obfuscation, packing, and polymorphism**. Malware authors employed techniques such as code packing, encryption, and frequent mutation to evade static signature detection. Although these techniques increased the complexity of analysis, malware artifacts still typically resided on disk, allowing for eventual detection through dynamic analysis and heuristic-based methods (Christodorescu and Jha, 2006).

The third era marked a significant shift toward **fileless malware**, where attackers minimized or entirely eliminated reliance on persistent files. Instead, malicious code was executed directly in memory, often through legitimate scripting engines or exploitation of system interpreters. This transition reduced the effectiveness of both static and dynamic file-based analysis, as malicious activity left few recoverable artifacts for post-incident investigation (Chen et al., 2014).

More recently, malware evolution has entered a fourth era dominated by **Living-off-the-Land (LOTL) attacks**. In this paradigm, attackers increasingly abuse pre-installed system tools, administrative utilities, and trusted binaries commonly referred to as Living-off-the-Land Binaries (LOLBins) to conduct malicious operations. Rather than deploying distinct malware payloads, adversaries orchestrate sequences of legitimate commands and processes to achieve execution, persistence, and command-and-control objectives. This approach allows malicious activity to blend seamlessly into normal system behavior, significantly complicating detection efforts (Harish and Swapna, 2024).

Across these eras, a clear trend emerges: malware has progressively shifted from **artifact-centric execution** toward **behavior-centric exploitation**. Each evolutionary step reflects an adaptation to prevailing defensive strategies, with attackers deliberately moving toward techniques that reduce observability and increase ambiguity. This progression underscores the limitations of traditional detection models and highlights the need for modern approaches capable of analyzing context, semantics, and inter-process relationships in real time.

2.3.3 Malware Classification in Modern Enterprise Environments

Malware classification has traditionally relied on observable characteristics such as file type, propagation method, and payload behavior. Early classification schemes categorized malware into distinct groups including viruses, worms, trojans, spyware, and ransomware, primarily based on file-based artifacts and infection vectors. While these taxonomies provided a useful foundation, academic research increasingly argues that such static classifications are insufficient for modern enterprise environments where malicious activity often lacks persistent artifacts and evolves dynamically (Khalid et al., 2023).

In contemporary enterprise settings, malware classification has shifted toward behavior-centric and context-aware perspectives. Rather than focusing solely on the presence of malicious binaries, modern classification approaches consider how system resources are utilized, how processes interact, and how execution patterns deviate from expected operational behavior. Studies on fileless and Living-off-the-Land (LOTL) attacks emphasize that malware frequently manifests as sequences of legitimate actions whose malicious nature emerges only through contextual correlation (Harish and Swapna, 2024).

One important dimension of modern malware classification is execution modality. Malware can be broadly categorized as file-based or fileless, depending on whether it relies on persistent artifacts stored on disk. Fileless malware executes primarily in memory or through native system utilities, significantly reducing its forensic footprint and evading traditional detection mechanisms. Research demonstrates that this distinction is particularly relevant in enterprise environments where administrative tools are widely deployed for legitimate purposes (Singh and Tripathy, 2025).

Another classification dimension involves the abuse of legitimate system components. LOTL malware leverages trusted binaries, scripting engines, and administrative frameworks such as PowerShell, Windows Management Instrumentation (WMI), and command interpreters to carry out malicious objectives. Academic studies note that such malware blurs the boundary between benign and malicious behavior, complicating classification efforts that rely on tool identity alone (Sudhakar and Kumar, 2020).

Modern malware classification also increasingly accounts for temporal and operational context. Instead of treating individual events as independent, researchers emphasize the importance of

analyzing execution sequences, process lineage, and timing relationships to distinguish malicious campaigns from routine administrative activity. Empirical studies highlight that contextual features such as process lifetime, invocation frequency, and parent-child relationships provide critical signals for identifying malicious behavior within enterprise workloads (Meher et al., 2025).

In summary, malware classification in modern enterprise environments extends beyond static labels and file-centric categories. Effective classification frameworks must incorporate behavioral patterns, execution context, and semantic characteristics to reflect the evolving nature of malware and to support robust detection and response strategies in complex operational settings.

2.3.4 Malware as a Behavioral Phenomenon

Recent advances in malware research increasingly conceptualize malware not as a static artifact, but as a behavioral phenomenon emerging from the interaction between system components, execution context, and temporal dynamics. This perspective reflects a growing consensus in academic literature that malicious activity is often indistinguishable from benign behavior when examined through isolated events or static features alone (Khalid et al., 2023).

In modern enterprise environments, malware frequently manifests through sequences of legitimate actions rather than overtly malicious binaries. Fileless and Living-off-the-Land (LOTL) attacks exemplify this shift by abusing trusted system utilities, scripting environments, and native execution mechanisms to achieve malicious objectives without introducing persistent artifacts. Studies emphasize that such attacks can only be reliably identified by analyzing execution behaviour over time, including process lineage, invocation patterns, and interaction structures (Singh and Tripathy, 2025).

A behavioural view of malware also highlights the importance of context in distinguishing malicious intent. Individual system events such as a PowerShell invocation or command-line execution may be benign in isolation yet become suspicious when correlated with preceding or subsequent actions. Research demonstrates that contextual signals, including execution timing, process duration, and parent-child relationships, are critical for differentiating malicious campaigns from routine administrative workflows (Ongun et al., 2021a).

Semantic characteristics further complement behavioural analysis. As attackers rely heavily on scripting languages and command interpreters, the semantic structure of commands and scripts provides valuable clues regarding intent. However, multiple studies caution that semantic analysis alone is insufficient, as benign and malicious scripts may share similar lexical and syntactic patterns. Effective malware characterization therefore requires the integration of semantic understanding with behavioural and contextual reasoning (Wu et al., 2022).

The behavioural perspective has significant implications for detection research. It challenges traditional malware models that assume the existence of discrete, identifiable artifacts and instead motivates approaches capable of correlating distributed signals across processes, time, and execution contexts. Empirical studies increasingly advocate for hybrid detection strategies that combine behavioural modelling with semantic analysis to address the stealthy and adaptive nature of modern malware.

2.4 Modern Malware Trends and the Emergence of Fileless Techniques

2.4.1 Decline of Traditional File-Based Malware

Traditional file-based malware, characterized by the deployment of malicious executable files stored on disk, has historically dominated the threat landscape. Detection strategies during this period relied heavily on static signatures, file hashes, and known binary patterns. While such approaches were effective against early malware families, academic research increasingly reports a steady decline in the relative prevalence and effectiveness of purely file-based attacks within modern enterprise environments (Khalid et al., 2023).

One primary factor contributing to this decline is the maturation of endpoint protection technologies. Advances in signature-based detection, heuristic scanning, and sandboxing have significantly raised the cost and reduced the success rate of deploying persistent malicious binaries. Empirical studies indicate that attackers have responded to these defensive improvements by shifting away from easily detectable file-based payloads toward more evasive techniques that minimize disk artifacts (Hendler et al., 2018).

Another contributing factor is the growing operational complexity of enterprise systems. Modern environments rely extensively on scripting engines, automation frameworks, and trusted administrative tools to manage large-scale infrastructures. Research highlights that this

operational reliance creates opportunities for attackers to bypass file-based delivery mechanisms altogether by abusing existing system functionality, thereby reducing the need for custom malware binaries (Meher et al., 2025).

Studies on malware evolution further demonstrate that file-based attacks are increasingly replaced or supplemented by in-memory execution, command-based payload delivery, and staged attack chains. These approaches allow adversaries to dynamically construct malicious behavior at runtime while evading traditional static inspection mechanisms. As a result, detection models that depend primarily on file artifacts face diminishing effectiveness in identifying contemporary threats.

Industry reports support these findings by documenting a measurable shift in attack techniques away from standalone malware binaries toward fileless and Living-off-the-Land (LOTL) methods. However, such reports primarily contextualize trends already established in academic research and should be interpreted as supplementary background rather than primary analytical evidence (CrowdStrike, 2024).

In summary, the decline of traditional file-based malware reflects both the success of legacy detection mechanisms and the adaptive evolution of adversarial techniques. This shift underscores the necessity for malware detection research to move beyond file-centric paradigms toward approaches that emphasize behavioral context, execution dynamics, and semantic analysis.

2.4.2 Rise of In-Memory and Script-Based Execution

The decline of traditional file-based malware has been accompanied by a significant rise in in-memory and script-based execution techniques. Rather than deploying persistent executable files, modern attackers increasingly execute malicious logic directly in memory or through interpreted scripts, significantly reducing their forensic footprint. Academic studies consistently report that this execution model enables adversaries to evade signature-based detection and file integrity monitoring mechanisms commonly deployed in enterprise environments.

In-memory execution allows malicious code to reside transiently within process memory without being written to disk. Techniques such as reflective DLL injection, process hollowing, and shellcode execution enable attackers to perform arbitrary actions while minimizing

observable artifacts. Research on fileless malware highlights that these techniques are particularly effective against endpoint security solutions that prioritize static or disk-based analysis (Korkin and Nesterov, n.d.).

Script-based execution further amplifies this evasion capability by leveraging trusted scripting engines and command interpreters already present in the operating system. PowerShell, Windows Management Instrumentation (WMI), and command-line interfaces are frequently abused to execute encoded commands, download payloads, or orchestrate multi-stage attacks. Empirical studies emphasize that such script-based attacks often appear indistinguishable from legitimate administrative activity when examined in isolation (Khalid et al., 2023).

A critical challenge posed by in-memory and script-based execution is the semantic and behavioral ambiguity it introduces. While script content may contain indicators of malicious intent, attackers commonly employ obfuscation techniques such as command encoding, string manipulation, and staged execution to evade static inspection. Research demonstrates that detecting these attacks requires analysis approaches capable of correlating semantic features with execution context, process lineage, and temporal behavior.

Industry threat intelligence reports corroborate academic findings by documenting a sustained increase in memory-resident and script-driven attack techniques across enterprise environments. However, as with earlier sections, such reports primarily contextualize trends identified in peer-reviewed research and should be treated as supplementary background rather than primary analytical evidence (Microsoft, 2023).

In summary, the rise of in-memory and script-based execution represents a fundamental shift in adversarial tradecraft. By minimizing persistent artifacts and abusing legitimate execution mechanisms, attackers challenge traditional detection paradigms and motivate the development of behaviour-aware and semantically informed detection strategies suitable for real-time enterprise deployment.

2.4.3 Abuse of Legitimate Administrative Tools (LOLBins)

Living-off-the-Land (LOTL) techniques refer to attack strategies in which adversaries leverage legitimate system tools and trusted binaries to perform malicious actions. Rather than introducing custom malware binaries, attackers abuse native utilities, scripting engines, and

administrative frameworks that are already present within the operating system. Academic literature emphasizes that LOTL techniques are particularly effective in enterprise environments, where such tools are widely used for routine administration and automation (Khalid et al., 2023).

Commonly abused LOTL components include PowerShell, Windows Management Instrumentation (WMI), command-line interpreters, scheduled task utilities, and native system binaries such as *rundll32*, *mshta*, and *regsvr32*. These tools provide attackers with powerful execution capabilities while operating under the guise of legitimate system activity. Studies highlight that malicious use of LOTL tools often exploits existing trust relationships and default execution permissions, enabling attackers to bypass application whitelisting and signature-based defenses (Harish and Swapna, 2024).

A defining challenge of LOTL-based attacks is their behavioural ambiguity. Because the same tools are routinely used by administrators, malware classification and detection approaches that rely on tool identity alone are insufficient. Research demonstrates that LOTL attacks frequently manifest through subtle deviations in execution patterns, such as unusual invocation sequences, abnormal parent-child relationships, or atypical execution timing, rather than overtly malicious commands (Ongun et al., 2021a).

LOTL techniques are commonly combined with in-memory execution and script-based payload delivery to further reduce forensic visibility. Attackers may download, decode, and execute payloads entirely within memory or orchestrate multi-stage attacks using chained commands across multiple tools. Empirical studies emphasize that such chaining behaviour complicates detection, as each individual action may appear benign in isolation while contributing to a malicious campaign when correlated over time (Leng et al., 2024).

Industry threat intelligence reports corroborate academic findings by documenting the widespread adoption of LOTL techniques across diverse threat actor groups, including APTs and financially motivated attackers. However, consistent with earlier sections, such reports primarily contextualize trends already identified in peer-reviewed research and should be treated as supplementary background rather than primary analytical sources (CrowdStrike, 2024).

In summary, LOTL techniques represent a significant evolution in adversarial tradecraft by enabling attackers to operate within trusted execution environments while minimizing detectable artifacts. This paradigm challenges traditional detection models and motivates research into behaviour-aware and context-sensitive approaches capable of identifying malicious misuse of legitimate tools in real time.

2.4.4 Increased Use of Obfuscation and Encoding Techniques

The increasing adoption of fileless and Living-off-the-Land (LOTL) attack strategies has been accompanied by a widespread use of obfuscation and encoding techniques. Rather than executing commands in clear and interpretable forms, modern malware frequently employs various transformation mechanisms to conceal intent, evade static inspection, and bypass rule-based detection systems. Academic studies consistently report that obfuscation has become a core component of contemporary malware tradecraft, particularly in script-based and in-memory attacks.

Encoding techniques such as Base64 encoding, string concatenation, and dynamic command reconstruction are commonly used to obscure malicious logic in scripting environments. PowerShell-based attacks, for example, often rely on encoded commands and staged execution to avoid detection by signature-based or keyword-driven security controls. Empirical research highlights that these techniques significantly reduce the effectiveness of static command analysis and simple pattern-matching approaches (Meher et al., 2025).

Beyond basic encoding, attackers increasingly employ advanced obfuscation strategies, including control-flow manipulation, variable renaming, dead-code insertion, and runtime deobfuscation. These techniques enable malicious scripts to adapt their execution behaviour dynamically, making it difficult for static analysis tools to accurately interpret intent prior to execution. Studies on fileless malware emphasize that such obfuscation techniques are particularly effective when combined with legitimate scripting engines and trusted system utilities (Sudhakar and Kumar, 2020).

A critical implication of widespread obfuscation is the semantic ambiguity it introduces into detection processes. While semantic analysis of scripts and commands can provide valuable insights, obfuscation often distorts lexical and syntactic features, causing benign and malicious scripts to appear structurally similar. Research demonstrates that effective detection of

obfuscated malware requires analysis approaches capable of correlating semantic signals with execution context, behavioural patterns, and temporal relationships (Leng et al., 2024)

Industry threat intelligence reports reinforce these findings by documenting a steady increase in the use of encoded and obfuscated payloads across enterprise-targeted attacks. However, consistent with earlier sections, such reports primarily serve to contextualize trends already established in academic research and should be interpreted as supplementary background rather than primary analytical evidence (CrowdStrike, 2024).

In summary, the increased use of obfuscation and encoding techniques represents a significant challenge for malware detection research. By concealing intent and minimizing static indicators, these techniques undermine traditional analysis methods and motivate the development of detection approaches that integrate semantic understanding with behavioural and contextual analysis.

2.4.5 Increased Focus on Stealth and Operational Blending

Detecting fileless malware presents a set of unique and persistent challenges that fundamentally differentiate it from traditional file-based threats. Because fileless attacks avoid creating persistent artifacts on disk, many conventional detection mechanisms such as signature matching, file reputation analysis, and static binary inspection are rendered ineffective. Academic research consistently reports that the absence of tangible malware artifacts significantly reduces the visibility of malicious activity within enterprise systems.

One major challenge arises from the heavy reliance of fileless malware on legitimate system tools and execution environments. Living-off-the-Land (LOTL) techniques exploit trusted binaries, scripting engines, and administrative frameworks that are commonly used for benign operational tasks. As a result, malicious executions often appear indistinguishable from legitimate administrative activity when examined in isolation. Studies emphasize that this behavioural overlap leads to high false positive rates when detection systems rely solely on tool identity or individual command features (Harish and Swapna, 2024).

Another critical challenge is the transient and dynamic nature of fileless malware execution. In-memory payloads and script-based commands may execute briefly and leave minimal forensic traces, limiting the effectiveness of post-incident analysis and retrospective detection.

Empirical studies highlight that attackers frequently exploit this temporal window to perform reconnaissance, lateral movement, and persistence establishment before detection mechanisms can respond (Khalid et al., 2023).

Obfuscation and encoding further complicate detection by distorting semantic indicators of malicious intent. Encoded commands, runtime obfuscation, and staged execution can mask malicious logic from static analysis tools and keyword-based detection rules. Research demonstrates that these techniques significantly reduce the reliability of purely semantic detection approaches, particularly when contextual execution information is unavailable (Elongha et al., 2025).

Scalability and real-time constraints also pose significant challenges for fileless malware detection. Enterprise environments generate vast volumes of telemetry data, requiring detection systems to operate efficiently under high throughput while maintaining low latency and false positive rates. Studies indicate that computationally intensive or offline analysis techniques may be impractical for deployment in production environments, despite their analytical capabilities.

In summary, the detection of fileless malware is challenged by the absence of disk artifacts, behavioural ambiguity, transient execution, obfuscation, and operational constraints. These challenges underscore the need for detection approaches that can correlate behavioural context, execution dynamics, and semantic information in real time to effectively identify malicious activity within complex enterprise systems.

2.4.6 Ineffectiveness of Traditional Detection Models

Traditional malware detection models were primarily designed under the assumption that malicious activity could be identified through static artifacts, well-defined signatures, or known behavioural indicators. These models typically rely on file-based analysis, rule matching, and predefined heuristics derived from previously observed attack patterns. While effective against early generations of malware, academic research increasingly demonstrates that such approaches are insufficient for detecting modern fileless and Living-off-the-Land (LOTL) attacks operating within enterprise environments (Khalid et al., 2023).

Signature-based detection mechanisms, in particular, struggle in scenarios where no persistent binaries are introduced to disk. Fileless malware and in-memory execution techniques bypass signature repositories entirely, rendering hash-based and pattern-matching systems ineffective. Studies consistently report that attackers deliberately exploit this limitation by constructing malicious behaviour dynamically at runtime, thereby avoiding detection by traditional antivirus and endpoint protection solutions (Khalid et al., 2023).

Rule-based and heuristic detection models face similar limitations. Although heuristic rules can capture known malicious behaviours, they are often brittle and prone to evasion through minor variations in execution patterns, command structures, or obfuscation techniques. Research highlights that attackers frequently adapt their tactics to evade static rules by encoding commands, fragmenting execution stages, or abusing legitimate tools in unconventional ways, resulting in high false negative rates (Carrier et al., 2022).

Another critical shortcoming of traditional detection models is their limited ability to reason over context and temporal relationships. Many legacy systems analyze events in isolation, failing to correlate process lineage, execution sequences, or long-term behavioural patterns. Empirical studies emphasize that modern attacks distribute malicious actions across multiple benign-looking events, each insufficient to trigger alerts independently, thereby exploiting the lack of contextual awareness in conventional detection pipelines (Ongun et al., 2021a).

Operational constraints further exacerbate these issues. Traditional detection models often require extensive offline analysis, manual rule tuning, or frequent signature updates to remain effective. In high-throughput enterprise environments, such approaches struggle to scale while maintaining acceptable detection latency and false positive rates. Research indicates that these limitations reduce the practical deployability of traditional models in real-time monitoring scenarios (Elongha et al., 2025).

In summary, the ineffectiveness of traditional detection models stems from their reliance on static artifacts, rigid rules, and limited contextual reasoning. As malware evolves toward fileless, stealth-driven, and behaviour-centric execution, detection research must move beyond legacy paradigms toward approaches capable of correlating semantic, behavioural, and temporal signals under real-time operational constraints.

2.4.7 Modern Attack Case Studies Illustrating Fileless Trends

Recent high-profile cyber incidents provide concrete evidence of the growing adoption of fileless and Living-off-the-Land (LOTL) techniques in real-world attacks. Academic analyses of modern enterprise intrusions reveal that many successful campaigns rely on legitimate system utilities, in-memory execution, and obfuscated scripts rather than deploying custom malware binaries. These case studies illustrate how theoretical trends discussed in previous sections manifest in operational environments (Khalid et al., 2023).

One frequently cited example involves the extensive abuse of PowerShell for post-exploitation activities. Research-based analyses document that attackers commonly leverage encoded PowerShell commands to perform reconnaissance, download secondary payloads, and establish persistence without writing executable files to disk. Such campaigns demonstrate how script-based execution and command obfuscation enable attackers to bypass traditional signature-based detection while maintaining flexibility and control over compromised systems.

Another representative case involves the misuse of Windows Management Instrumentation (WMI) and scheduled tasks to achieve persistence and lateral movement. Academic case studies show that attackers can deploy fileless backdoors by registering malicious WMI event consumers or task schedulers that execute benign-looking commands at predefined intervals. These techniques exemplify how attackers exploit trusted administrative frameworks to blend malicious activity into normal operational workflows (Berrios et al., 2025).

Large-scale enterprise intrusions have also demonstrated the effectiveness of LOTL chaining techniques, where attackers combine multiple legitimate tools such as *cmd.exe*, *rundll32*, *mshta*, and scripting engines into multi-stage execution sequences. Empirical studies highlight that each individual command in such chains may appear benign, yet their collective behavior reveals malicious intent when analyzed through process lineage and execution context (Ongun et al., 2021a).

Industry threat reports frequently reference prominent attack campaigns that align with these academic observations, including intrusions attributed to advanced persistent threat (APT) groups and ransomware operators. While industry reports provide valuable operational context, they largely reinforce patterns already identified in peer-reviewed research and should therefore be interpreted as illustrative rather than primary analytical sources (Microsoft, 2023).

In summary, modern attack case studies consistently demonstrate the practical viability and effectiveness of fileless and LOTL techniques in enterprise environments. These cases highlight the limitations of traditional detection models and reinforce the need for research approaches that emphasize behavioral correlation, contextual reasoning, and semantic analysis to identify malicious activity embedded within legitimate system operations.

2.4.8 Implications for Detection Research

The evolution of modern malware toward fileless execution, Living-off-the-Land (LOTL) techniques, and extensive obfuscation has profound implications for malware detection research. Collectively, the trends discussed in Sections 2.4.1 through 2.4.7 demonstrate that adversaries increasingly operate within legitimate system boundaries, rendering traditional detection assumptions such as the presence of malicious files or clearly distinguishable indicators largely obsolete (Khalid et al., 2023).

One major implication is the diminished effectiveness of artifact-centric detection approaches. As attackers avoid persistent binaries and rely on in-memory execution and script-based payloads, detection research must move beyond file-level analysis toward methods capable of reasoning over execution behavior and system interactions. Academic studies consistently emphasize that effective detection increasingly depends on identifying anomalous behavioral patterns rather than static indicators of compromise.

Another critical implication concerns the role of contextual awareness in detection systems. Modern attacks often distribute malicious activity across multiple benign-looking actions, each insufficient to trigger alerts when analyzed in isolation. Research highlights that correlating execution context such as process lineage, execution timing, and interaction relationships is essential for uncovering malicious intent embedded within legitimate operational workflows (Harish and Swapna, 2024).

The widespread use of obfuscation and encoding further underscores the limitations of purely semantic or rule-based detection techniques. While semantic analysis of scripts and commands provides valuable information, obfuscation can distort lexical and syntactic features, reducing detection reliability. Empirical studies therefore advocate detection approaches that integrate semantic understanding with behavioral and contextual analysis to improve robustness against evasive tactics.

Operational constraints also shape detection research priorities. Enterprise environments require detection systems that can operate in real time, scale to high event volumes, and maintain low false positive rates. Research indicates that detection models must balance analytical sophistication with computational efficiency to remain deployable in production settings, particularly for endpoint and log-based monitoring systems (Elongha et al., 2025).

In summary, modern malware trends necessitate a fundamental shift in detection research toward integrated, behavior-aware, and context-sensitive approaches. Detection systems must be capable of correlating semantic, behavioral, and temporal signals under real-time constraints to effectively identify stealthy fileless attacks within complex enterprise environments. These implications motivate the exploration of hybrid detection frameworks discussed in subsequent chapters.

2.5 Fileless and Living-off-the-Land (LOTL) Attacks

Fileless and Living-off-the-Land (LOTL) attacks represent a significant evolution in modern malware operations, characterized by the absence of persistent malicious binaries and the extensive abuse of legitimate system tools. Unlike traditional malware that relies on executable files stored on disk, fileless attacks execute malicious logic directly in memory or through trusted operating system components, thereby minimizing detectable artifacts and evading conventional security controls (Khalid et al., 2023).

Academic literature emphasizes that LOTL attacks exploit the dual-use nature of native system utilities, scripting engines, and administrative frameworks that are essential for enterprise operations. Tools such as PowerShell, Windows Management Instrumentation (WMI), command interpreters, and scheduled task utilities provide attackers with powerful capabilities while operating under the guise of legitimate activity. This abuse fundamentally blurs the distinction between benign and malicious behaviour, complicating both malware classification and detection (Harish and Swapna, 2024).

Fileless and LOTL attacks are frequently employed as part of multi-stage intrusion campaigns rather than standalone exploits. Initial access may be achieved through phishing, credential abuse, or vulnerability exploitation, followed by post-exploitation activities that rely heavily on in-memory execution and chained LOTL commands. Empirical studies demonstrate that such

campaigns emphasize stealth, persistence, and adaptability, allowing attackers to maintain long-term presence within enterprise environments while avoiding detection.

A defining challenge of fileless and LOTL attacks lies in their behavioural and contextual nature. Individual actions such as script execution or command invocation may appear benign when examined in isolation yet collectively form malicious execution patterns. Research consistently highlights that effective detection of these attacks requires correlating execution context, process relationships, and temporal behaviour rather than relying on static indicators or isolated events.

From a detection research perspective, fileless and LOTL attacks expose fundamental limitations in traditional security models. Signature-based, rule-based, and artifact-centric approaches struggle to identify malicious behaviour embedded within legitimate system operations. Consequently, academic research increasingly advocates for detection frameworks that integrate behavioural analysis, contextual reasoning, and semantic interpretation to address the stealthy and adaptive characteristics of these attacks (Leng et al., 2024).

In summary, fileless and Living-off-the-Land attacks represent a paradigm shift in adversarial tradecraft, emphasizing behaviour over artifacts and stealth over visibility. Understanding their characteristics, execution patterns, and detection challenges is essential for developing effective malware detection approaches tailored to modern enterprise environments. The following sections examine existing detection strategies and research gaps related to fileless and LOTL attacks in greater detail.

2.5.1 Definition and Characteristics of Fileless Malware

Fileless malware refers to a class of malicious attacks that operate without relying on persistent executable files stored on disk. Instead of deploying traditional malware binaries, fileless attacks execute malicious logic directly in system memory or through trusted operating system components such as scripting engines, command interpreters, and administrative utilities. Academic literature emphasizes that this execution model significantly reduces the presence of detectable artifacts, enabling attackers to evade traditional file-based security controls (Khalid et al., 2023).

A defining characteristic of fileless malware is its heavy reliance on legitimate system functionality. Fileless attacks commonly abuse tools such as PowerShell, Windows Management Instrumentation (WMI), and native command-line utilities to perform actions including code execution, payload delivery, persistence, and lateral movement. Studies highlight that this abuse exploits the dual-use nature of administrative tools, which are essential for enterprise operations but can be repurposed for malicious objectives (Meher et al., 2025).

Another key characteristic of fileless malware is its transient and dynamic execution behaviour. Malicious payloads may exist only briefly in memory, execute in stages, or be reconstructed dynamically at runtime. This ephemeral execution significantly complicates forensic analysis and retrospective detection, as little to no persistent evidence may remain after execution. Empirical research demonstrates that attackers often leverage this transient behaviour to minimize exposure and maximize dwell time within compromised environments (Leng et al., 2024).

Fileless malware is also closely associated with obfuscation and encoding techniques. To conceal malicious intent, attackers frequently employ encoded commands, script obfuscation, and staged execution mechanisms that hinder static analysis. Research indicates that these techniques are particularly effective in fileless contexts, where semantic indicators may be distorted or dynamically generated during execution (Ongun et al., 2021a).

From a behavioural perspective, fileless malware is best understood as a sequence of actions rather than a discrete object. Individual execution events may appear benign in isolation yet collectively form malicious patterns when correlated across time and process relationships. Studies emphasize that effective detection of fileless malware requires contextual and temporal analysis capable of capturing these behavioural patterns.

In summary, fileless malware is characterized by the absence of persistent disk artifacts, the abuse of legitimate system tools, transient in-memory execution, and extensive use of obfuscation. These characteristics fundamentally challenge traditional detection models and motivate the development of behaviour-aware and context-sensitive detection approaches tailored to modern enterprise environments.

2.5.2 Living-off-the-Land (LOTL) Attack Techniques

Living-off-the-Land (LOTL) attack techniques involve the malicious exploitation of legitimate system tools and native binaries to perform actions traditionally associated with malware. Instead of deploying custom executables, attackers rely on trusted operating system components that are already present in enterprise environments. Academic research emphasizes that this strategy significantly reduces the attack surface visible to traditional detection mechanisms while increasing the difficulty of distinguishing malicious activity from benign administrative behavior (Khalid et al., 2023).

A common LOTL technique is the abuse of scripting engines and command interpreters for execution and orchestration. PowerShell, command-line interfaces, and Windows scripting environments are frequently leveraged to execute encoded commands, retrieve payloads, and coordinate multi-stage attacks. Studies highlight that such executions often appear syntactically valid and operationally legitimate, complicating detection efforts that rely on static command inspection or tool identity (Ongun et al., 2021a).

Another prevalent LOTL technique involves the misuse of system management and automation frameworks to establish persistence and lateral movement. Windows Management Instrumentation (WMI), scheduled tasks, and registry-based mechanisms can be abused to trigger malicious actions under normal system workflows. Empirical studies demonstrate that these techniques allow attackers to maintain long-term access without deploying persistent malware binaries, thereby evading signature-based and file-centric defenses (Leng et al., 2024).

LOTL attacks also frequently employ chaining techniques, in which multiple legitimate tools are combined into coordinated execution sequences. For example, attackers may use a command interpreter to invoke a scripting engine, which in turn loads payloads into memory or triggers additional system utilities. Research emphasizes that each individual command in such chains may appear benign, yet the overall execution pattern reveals malicious intent when analyzed through process lineage and temporal correlation (Chidambaram and M P, 2025).

Obfuscation and encoding are integral to many LOTL techniques. Attackers often encode commands, dynamically reconstruct scripts, or fragment execution stages to evade static detection and rule-based systems. Studies indicate that these techniques further blur the

boundary between benign and malicious activity, reinforcing the need for detection approaches that integrate semantic analysis with behavioral and contextual reasoning.

In summary, LOTL attack techniques exploit trusted system components, execution chaining, and obfuscation to achieve stealth and persistence within enterprise environments. These techniques challenge traditional detection models and motivate research into behavior-aware and context-sensitive detection strategies capable of identifying malicious misuse of legitimate tools in real time.

2.5.3 Fileless/LOTL Attack Lifecycle (Detailed Technical Analysis)

Fileless and Living-off-the-Land (LOTL) attacks generally unfold through a structured, multi-stage lifecycle that integrates conventional intrusion phases with advanced, stealth-oriented post-exploitation strategies. In contrast to traditional file-based malware campaigns where malicious binaries are written to disk and leave tangible forensic artifacts fileless and LOTL operations deliberately minimize persistent footprints by operating primarily in memory and by leveraging legitimate system tools and administrative utilities. This strategic reliance on trusted, native components (e.g., scripting engines, system management frameworks, and command-line interpreters) enables adversaries to blend malicious activities with normal system operations, thereby complicating attribution and detection.

The lifecycle of such attacks commonly encompasses stages including initial access, execution, persistence (often memory-resident or registry-based), privilege escalation, lateral movement, command-and-control communication, and objective fulfilment. However, what differentiates fileless/LOTL campaigns is the systematic emphasis on evasion throughout each stage. Rather than introducing foreign executables, attackers repurpose built-in functionalities to execute payloads, deliver commands, and maintain operational continuity without triggering signature-based defenses. This approach shifts the observable indicators from static artifacts to behavioural and semantic anomalies embedded within legitimate system processes.

Scholarly research consistently highlights that a comprehensive understanding of this lifecycle structure is fundamental for the development of robust detection and response mechanisms. Each phase produces distinct behavioural patterns, telemetry traces, and contextual signals ranging from anomalous process invocation chains and unusual command-line arguments to irregular parent-child process relationships and memory execution patterns. Consequently,

effective defensive strategies require phase-aware monitoring frameworks capable of correlating multi-stage behavioral evidence rather than relying solely on artifact-based detection. By conceptualizing fileless and LOTL attacks as dynamic, lifecycle-driven phenomena, researchers and practitioners can design more resilient analytical models that address both technical evasion tactics and the broader operational logic underlying these threats.

Phase 1: Initial Access and Execution Trigger

The lifecycle commonly begins with an initial access vector such as phishing, credential abuse, or exploitation of exposed services. The key distinction for fileless/LOTL attacks is that the initial payload is often lightweight and designed to initiate script-based or in-memory execution rather than drop a persistent binary. Research highlights that early-stage execution may involve command interpreters or scripting engines to bootstrap subsequent actions while maintaining a low forensic footprint (Khalid et al., 2023).

Detection implications: early-stage signals may include atypical command invocations, unusual parent processes, or suspicious launch timing relative to user activity.

Phase 2: Payload Retrieval, Decoding, and In-Memory Staging

In many fileless attacks, malicious logic is retrieved dynamically (e.g., downloaded, extracted, or reconstructed) and prepared for execution in memory. This phase often involves decoding or deobfuscation mechanisms such as Base64 decoding, string reconstruction, or staged script execution. Academic research emphasizes that this stage is critical for evasion, as the payload may never exist as a stable on-disk artifact (Joshi and Warrier, 2025).

Detection implications: semantic indicators may appear in command structures (e.g., encoded parameters), but robust detection typically requires linking these semantics with contextual signals such as process ancestry and execution sequences.

Phase 3: Persistence via Legitimate System Mechanisms

Fileless/LOTL attacks commonly establish persistence without dropping binaries by abusing legitimate OS mechanisms. Techniques may include scheduled tasks, WMI event subscriptions, registry-based execution triggers, or script-based autoruns. Studies demonstrate that these methods exploit trusted system administration features, allowing malicious persistence to resemble normal operational configurations (Khalid et al., 2023)

Detection implications: persistence signals often emerge from abnormal configuration changes, unusual process service interactions, or repeated execution patterns over time, rather than a new file creation event.

Phase 4: Privilege Escalation and Credential Access

After persistence is established, attackers often escalate privileges and obtain credentials to expand control. In fileless/LOTL settings, these actions may be conducted using native tools, memory-based credential harvesting, or abuse of authentication processes. Empirical studies highlight that the stealth advantage of LOTL tools enables attackers to perform credential-related actions while blending into routine administrative behavior (Chidambaram and M P, 2025).

Detection implications: behaviour-based features such as abnormal process lineage, unusual access patterns, and anomalous execution duration may provide stronger signals than static command content alone.

Phase 5: Lateral Movement and Command Execution Chaining

LOTL attacks frequently depend on lateral movement through remote administration interfaces, scripting frameworks, or chained execution across multiple tools. Attackers may coordinate sequences such as *cmd* → *PowerShell* → *in-memory loader* → *system utility*, distributing malicious logic across multiple steps. Research emphasizes that such chaining behavior is difficult to detect using isolated events, and requires correlation across process relationships and temporal sequences (Ongun et al., 2021a).

Detection implications: graph-like process relationships (parent-child chains, degrees, and timing) become valuable for modelling execution context and identifying suspicious propagation patterns.

Phase 6: Actions on Objectives and Cleanup

In the final stage, attackers execute objectives such as data exfiltration, sabotage, encryption (ransomware), or long-term surveillance. Fileless and LOTL actors may also attempt to remove traces, disable logs, or blend activity into normal workflows. Studies note that evidence at this stage can be fragmented across logs and process telemetry, reinforcing the importance of early detection and contextual correlation (Khalid et al., 2023).

Detection implications: late-stage indicators may be visible through anomalous outbound connections or abnormal tool usage frequency, but effective mitigation often depends on detecting earlier lifecycle phases before impact occurs.

In summary, the fileless/LOTL attack lifecycle is characterized by staged execution, in-memory payload handling, abuse of legitimate persistence mechanisms, and chained tool-based actions. This lifecycle structure highlights why detection research must incorporate behavioral context, semantic understanding, and temporal correlation to identify attacks that deliberately avoid static artifacts and overt indicators.

2.5.4 Common Fileless/LOTL Techniques and LOLBin Abuse

Common fileless and Living-off-the-Land (LOTL) techniques revolve around the abuse of legitimate operating system binaries commonly referred to as Living-off-the-Land Binaries (LOLBins) to perform malicious actions. Rather than introducing custom malware tools, attackers leverage trusted executables and scripting environments that are natively available within enterprise systems. Academic research emphasizes that the widespread availability and legitimate use of LOLBins make them particularly attractive for stealthy attack campaigns (Khalid et al., 2023).

One prevalent technique involves the use of scripting engines for command execution and payload orchestration. PowerShell-based attacks frequently employ encoded commands, remote script retrieval, and in-memory execution to initiate and control malicious activity. Studies highlight that these commands often appear syntactically valid and operationally legitimate, complicating detection approaches that rely on static command inspection or keyword-based rules (Ongun et al., 2021a).

Another common technique is the abuse of native binaries for proxy execution. Tools such as *rundll32*, *mshta*, *regsvr32*, and *wmic* can be leveraged to execute malicious code indirectly by loading scripts, dynamic libraries, or remote content. Research demonstrates that such proxy execution techniques allow attackers to bypass application whitelisting and security controls that focus on executable provenance rather than execution context (Uroz and Rodríguez, 2020).

Persistence is frequently achieved through LOLBin abuse without introducing new binaries. Scheduled tasks, WMI event subscriptions, registry-based autoruns, and service configuration

changes can be manipulated to trigger malicious actions using legitimate executables. Empirical studies indicate that these techniques enable long-term persistence while maintaining a low forensic footprint, as the underlying mechanisms are commonly used for routine system management (Leng et al., 2024).

LOTL techniques are often combined with execution chaining and staged payload delivery. Attackers may orchestrate sequences in which multiple LOLBins interact across different phases of the attack lifecycle, distributing malicious logic across several benign-looking steps. Research emphasizes that such chaining behavior is difficult to detect when events are analyzed independently and requires correlation across process relationships, execution timing, and command semantics (Chidambaram and M P, 2025).

From a detection perspective, LOLBin abuse introduces significant ambiguity. Because the same binaries and commands are routinely used for legitimate purposes, detection approaches that rely solely on tool identity or static signatures are insufficient. Academic literature increasingly advocates for behavior-aware detection strategies that analyze how, when, and in what context LOLBins are used, rather than whether they are used at all.

In summary, common fileless and LOTL techniques exploit trusted system binaries, proxy execution mechanisms, and legitimate persistence frameworks to achieve stealth and resilience. Understanding these abuse patterns is critical for developing detection approaches capable of distinguishing malicious misuse of LOLBins from normal enterprise operations.

Prior to examining the table, it is necessary to emphasize that the abuse of command and script interpreters represents one of the most prevalent Living-off-the-Land (LOTL) execution mechanisms observed in contemporary cyber intrusions. Owing to the inherent legitimacy and widespread administrative use of utilities such as PowerShell and the Windows Command Shell, adversaries are able to execute malicious payloads while minimizing reliance on externally introduced binaries, thereby reducing their observable footprint and increasing operational stealth. This dual-use nature complicates detection efforts, as the same execution behaviors may also arise from routine system administration and enterprise automation workflows. Accordingly, the table below synthesizes common interpreter abuse patterns, their behavioral characteristics, and corresponding MITRE ATT&CK mappings to facilitate systematic analysis and more robust detection strategies.

1. Command & Script Interpreter Abuse (Execution)

Table 2.1 Script Interpreter Abuse (execution)

LOTL Pattern	Behavioral Description	MITRE ID	Technique	Tactic
Encoded PowerShell	powershell -enc Base64 payload	T1059.001	PowerShell	Execution
Hidden PowerShell Window	-w hidden	T1059.001	PowerShell	Execution
PowerShell No Profile	-nop to avoid profile logging	T1059.001	PowerShell	Execution
Command Prompt Abuse	cmd.exe /c chained execution	T1059.003	Windows Command Shell	Execution
WMI Command Execution	wmic process call create	T1047	Windows Management Instrumentation	Execution
MSHTA Script Execution	mshta.exe http://...	T1218.005	MSHTA	Execution
Rundll32 Abuse	rundll32.exe javascript:	T1218.011	Rundll32	Execution

Table 2.1 summarizes common script interpreter abuse patterns associated with Living-off-the-Land (LOTL) execution techniques, as mapped to the MITRE ATT&CK framework. The table highlights how native Windows interpreters and utilities such as PowerShell, Command Prompt, Windows Management Instrumentation (WMI), MSHTA, and Rundll32 can be leveraged to execute malicious logic without introducing new executable files to the system.

Several of the listed patterns, including encoded PowerShell commands, hidden execution windows, and the suppression of profile logging, are frequently used to evade user awareness and traditional security controls. Although these behaviours rely on legitimate system tools, their execution characteristics often deviate from normal operational usage when analyzed in context.

This overview motivates the need for detection mechanisms that go beyond file-based indicators. In particular, it underscores the importance of analyzing command-line semantics and execution context such as process lineage and interpreter invocation patterns which form the foundation of the proposed context-semantic detection framework presented in this study.

2. Defense Evasion (Fileless & Obfuscation)

Table 2.2 Defense Evasion (Fileless & Obfuscation)

LOTL Pattern	Description	MITRE ID	Technique	Tactic
Base64 Encoding	Encoded payload in command line	T1027	Obfuscated Files or Information	Defense Evasion
Living-off-the-Land Binary	Use of trusted binaries	T1218	Signed Binary Proxy Execution	Defense Evasion
Disable AMSI	AmsiUtils reflection	T1562.001	Disable Security Tools	Defense Evasion
Hidden Execution	No visible UI artifacts	T1564.003	Hidden Window	Defense Evasion
In-Memory Execution	No file dropped to disk	T1055	Process Injection (<i>conceptual</i>)	Defense Evasion

Table 2.2 summarizes common defense evasion techniques employed in fileless and Living-off-the-Land (LOTL) attacks, as mapped to the MITRE ATT&CK framework. The table highlights how adversaries leverage obfuscation, trusted system binaries, and in-memory execution to bypass traditional security mechanisms while maintaining a minimal on-disk footprint.

Techniques such as Base64-encoded command-line payloads and hidden execution windows are frequently used to conceal malicious intent from both users and signature-based detection systems. In addition, the abuse of trusted binaries and in-memory execution allows attackers to evade file-based scanning and exploit legitimate system functionality. The inclusion of AMSI bypass techniques further demonstrates how attackers may neutralize security controls designed to inspect script content at runtime.

This overview reinforces the need for detection approaches that analyze execution semantics and behavioral context rather than relying solely on static indicators. By explicitly mapping these evasion techniques to MITRE ATT&CK identifiers, the table provides a structured foundation for understanding how the proposed context–semantic detection framework targets stealthy and obfuscated fileless attack behaviors.

3. Parent-Child Process Abuse (Behavioral Context)

Table 2.3 Parrent-Child Process Abuse

Pattern	Description	MITRE ID	Technique	Tactic
Explorer → PowerShell	User-context abuse	T1204	User Execution	Execution
Office → PowerShell	Macro-style behavior	T1204.002	Malicious File	Execution
Svchost → Script Engine	System abuse	T1059	Command Interpreter	Execution
Service → PowerShell	Privileged execution	T1543	Create or Modify System Process	Persistence

Table 2.3 summarizes common parent–child process abuse patterns observed in fileless and Living-off-the-Land (LOTL) attack scenarios, as mapped to the MITRE ATT&CK framework. The table highlights how adversaries exploit abnormal process lineage relationships to execute malicious actions while blending into legitimate system behavior.

Patterns such as *explorer.exe* spawning *powershell.exe* or Office applications launching script interpreters are frequently associated with user-driven or document-based attack vectors. Similarly, the execution of scripting engines from system processes such as *svchost.exe* represents higher-privilege abuse that deviates from expected execution flows. These parent–child relationships are often difficult to detect using signature-based methods because the involved executables themselves are legitimate.

This overview emphasizes the importance of contextual analysis in fileless attack detection. By modeling parent–child process relationships as contextual features, the proposed framework

can identify anomalous execution patterns that indicate potential abuse, even in the absence of malicious binaries or explicit indicators.

4. Discovery & Reconnaissance via LOTL

Table 2.4 Discovery & Reconnaissance

Pattern	Description	MITRE ID	Technique	Tactic
System Info	systeminfo, whoami	T1082	System Information Discovery	Discovery
Process Enumeration	tasklist, Get-Process	T1057	Process Discovery	Discovery
Network Discovery	ipconfig, netstat	T1016	Network Discovery	Discovery
Account Discovery	net user	T1087	Account Discovery	Discovery

Table 2.4 summarizes common discovery and reconnaissance activities associated with fileless and Living-off-the-Land (LOTL) attack campaigns, as mapped to the MITRE ATT&CK framework. The table highlights how adversaries leverage built-in Windows commands and utilities to gather system, process, network, and account information without deploying additional tools or binaries.

Commands such as *systeminfo*, *whoami*, *tasklist*, *ipconfig*, and *netstat* are frequently used during the early stages of an attack to profile the target environment, identify running processes, enumerate network configuration, and assess user privileges. Because these commands are also commonly used for legitimate administrative purposes, their malicious intent is often difficult to infer from isolated execution events.

This overview reinforces the importance of contextual and semantic analysis for detecting reconnaissance activities. By correlating discovery commands with execution context, frequency, parent-child process relationships, and subsequent attack stages, the proposed framework can distinguish routine administrative usage from reconnaissance behaviour indicative of fileless and LOTL attacks.

5. Persistence via LOTL

Table 2.5 Persistence via LOTL

Pattern	Description	MITRE ID	Technique	Tactic
Registry Run Key	PowerShell modifying registry	T1547.001	Registry Run Keys	Persistence
Scheduled Task	schtasks /create	T1053.005	Scheduled Task	Persistence
WMI Event Subscription	Fileless persistence	T1546.003	WMI Event Subscription	Persistence

Table 2.5 summarizes persistence-related Living-off-the-Land (LOTL) techniques mapped to the MITRE ATT&CK framework. The table highlights how common Windows-native mechanisms such as registry run keys, scheduled tasks, and WMI event subscriptions can be abused to establish persistence without deploying traditional malicious binaries.

This mapping is important because it motivates the proposed framework's focus on monitoring process execution context and command-line semantics. In particular, persistence behaviours implemented through native tools often appear similar to legitimate administrative activity; therefore, effective detection requires contextual correlation (e.g., parent-child process lineage) and semantic cues (e.g., suspicious PowerShell or task scheduling parameters) rather than file-based indicators alone.

6. Command & Control (Fileless Contextual)

Table 2.6 Command & Control

Pattern	Description	MITRE ID	Technique	Tactic
Script Download	Invoke-WebRequest	T1105	Ingress Tool Transfer	C2
Living-off-the-Net	Using built-in networking	T1071.001	Web Protocols	C2

While the individual techniques shown in Figure 2.6 are well documented in the MITRE ATT&CK framework, their significance in LOTL detection lies in how they form correlated behavioral relationships rather than isolated indicators. In this study, such relationships are explicitly modeled as graph structures, where processes, commands, and system components

are represented as nodes, and their interactions are captured as edges. Table 2.7 summarizes the top-ranked graph nodes and edges most frequently associated with LOTL attacks, as learned by the proposed detection framework.

Table 2.7 presents the **Top 20 Graph Nodes and Edges Associated with LOTL Attacks**, derived from an ATT&CK-driven behavioral dataset used in this research.

Table 2.7 Top 20 Graph Nodes and Edges Associated with LOTL Attacks

Rank	Node (Entity)	Edge (Relation)	Behavioral Interpretation	Security Implication
1	powershell.exe	spawned_by (winword.exe)	Script execution via document	LOTL execution
2	powershell.exe	uses_encoded_command	Obfuscated execution	Evasion technique
3	cmd.exe	spawned_by (wmiprvse.exe)	WMI-based execution	Fileless persistence
4	rundll32.exe	executes_remote_dll	DLL abuse	LOLBin abuse
5	mshta.exe	loads_remote_script	HTA-based execution	Initial access
6	powershell.exe	downloads_payload	Ingress tool transfer	C2 preparation
7	schtasks.exe	creates_scheduled_task	Persistence mechanism	Privilege abuse
8	regsvr32.exe	executes_remote_script	Living-off-the-land execution	Defense evasion
9	wmic.exe	executes_process	Lateral execution	LOTL orchestration
10	winword.exe	spawns_powershell	Document-based attack	User-assisted execution
11	powershell.exe	modifies_registry	Registry-based persistence	Stealth persistence
12	powershell.exe	disables_security_feature	AMSI bypass attempt	Defense evasion
13	services.exe	spawns_powershell	Privileged execution	Persistence
14	svchost.exe	spawns_script_engine	System service abuse	Privilege escalation

15	powershell.exe	executes_in_memory	Fileless execution	Anti-forensics
16	cmd.exe	chains_multiple_commands	Command chaining	Obfuscation
17	powershell.exe	invokes_web_request	Network-based payload retrieval	C2 communication
18	taskeng.exe	executes_scheduled_payload	Task-based persistence	Stealth execution
19	explorer.exe	spawns_powershell	User-context execution	Social engineering vector
20	powershell.exe	executes_reflective_code	Memory-only execution	Advanced evasion

The graph nodes and edges presented in Table 2.7 are **not manually defined detection rules**, but represent dominant behavioral relations **learned from the ATT&CK-aligned dataset** and modeled through the proposed graph neural network. These relations emerge naturally from observed system telemetry and reflect how LOTL attacks orchestrate execution, evasion, and persistence by chaining legitimate system components.

By explicitly modeling these node–edge relationships, the proposed framework captures **behavioral context and inter-process dependencies** that are often missed by feature-centric or rule-based approaches. This graph-based representation also provides a foundation for downstream explainability analysis, enabling analysts to trace high-risk predictions back to interpretable behavioral paths.

From an analytical perspective, this chained behavior naturally lends itself to **graph-based representations**, where processes, commands, and system entities can be modeled as nodes, and their interactions as edges. Identifying recurring LOTL patterns within such graphs enables the extraction of dominant nodes and edges associated with malicious behavior providing a foundation for both detection and explainability. This observation motivates the use of graph-aware analytical models in subsequent sections of this thesis.

In summary, common fileless and LOTL techniques revolve around the systematic abuse of trusted binaries, scripting engines, and native system services. Their effectiveness lies not in exploiting novel vulnerabilities, but in orchestrating legitimate functionality into malicious behavioral patterns. Understanding these techniques is essential for designing detection

frameworks capable of identifying malicious intent embedded within normal operational workflows.

2.5.5 Why Fileless and LOTL Attacks Are Difficult to Detect

Fileless and Living-off-the-Land (LOTL) attacks pose significant detection challenges due to their fundamental reliance on legitimate system functionality, contextual ambiguity, and minimal forensic footprint. Unlike traditional malware, which introduces identifiable malicious artifacts, fileless attacks deliberately operate within trusted execution environments, making it difficult to distinguish malicious behavior from normal operational activity.

One of the primary challenges arises from the **absence of persistent artifacts**. Fileless attacks often execute entirely in memory or through transient scripts, leaving little to no evidence on disk once execution completes. This characteristic renders traditional file-scanning and signature-based detection mechanisms largely ineffective, as there are no stable artifacts to analyze or quarantine (Kara, 2023).

Another critical challenge is the **abuse of trusted system tools and binaries**. LOTL attacks leverage administrative utilities such as PowerShell, WMI, and scheduled task services components that are integral to enterprise operations. Since these tools are frequently used by system administrators and automated processes, their presence alone cannot be treated as suspicious. Detection systems must therefore infer malicious intent from execution context and behavioral patterns rather than from tool usage itself (Intelligence, 2022).

Fileless and LOTL attacks also exhibit **high behavioral overlap with benign activity**. Individual actions such as command execution, process spawning, or registry modification may appear entirely legitimate when analyzed in isolation. Malicious intent often emerges only when these actions are correlated over time and across system components. This temporal and relational complexity significantly complicates detection efforts that rely on isolated event analysis or short observation windows.

Obfuscation and encoding techniques further exacerbate detection difficulty. Encoded commands, dynamic script generation, and runtime payload decryption conceal malicious logic and undermine detection methods that rely on syntactic inspection or keyword matching.

Without semantic interpretation, detection systems struggle to differentiate between benign automation scripts and malicious command execution (Kara, 2023).

Operational constraints also contribute to the challenge. Many advanced detection approaches demonstrate strong performance in offline analysis but impose substantial computational overhead, making them unsuitable for real-time deployment. Fileless attacks frequently execute rapidly and terminate within seconds, requiring detection systems to operate under strict latency constraints. Delayed detection may occur after the malicious objective has already been achieved, reducing practical effectiveness.

Finally, the **evolving and adaptive nature** of LOTL techniques limits the longevity of static detection rules. Attackers continuously modify command syntax, execution paths, and tool combinations to evade known detection logic. This adaptability necessitates detection models that can generalize across unseen behaviors while maintaining low false positive rates in dynamic enterprise environments.

In summary, fileless and LOTL attacks are difficult to detect due to their lack of persistent artifacts, reliance on trusted system tools, behavioral overlap with benign activity, use of obfuscation, and real-time execution constraints. These challenges highlight the limitations of traditional detection models and underscore the need for context-aware, behavior-driven, and explainable detection frameworks.

These detection challenges directly motivate the development of advanced analytical frameworks capable of capturing behavioral context, semantic intent, and relational structure, which is discussed in the following subsection.

2.5.6 Implications for Detection Research

The characteristics and challenges associated with fileless and Living-off-the-Land (LOTL) attacks carry important implications for malware detection research. As demonstrated in the preceding sections, modern attacks increasingly avoid persistent artifacts, exploit legitimate system tools, and distribute malicious behaviour across multiple execution stages. These properties fundamentally undermine traditional detection assumptions and necessitate a rethinking of how malicious activity is modelled and identified within enterprise environments (Khalid et al., 2023).

One key implication is the need to shift from artifact-centric detection toward behaviour-centric analysis. Because fileless attacks do not rely on identifiable binaries, detection research must focus on execution patterns, process interactions, and deviations from expected operational behaviour. Academic studies consistently emphasize that understanding *how* system components are used over time is more informative than analyzing isolated events or static indicators.

Another implication concerns the importance of contextual correlation. Fileless and LOTL attacks often manifest as sequences of benign-looking actions whose malicious intent emerges only when execution context such as process lineage, execution timing, and interaction relationships is considered. Research highlights that detection systems lacking contextual awareness are prone to both false negatives and false positives, underscoring the necessity of correlating behavioural signals across processes and time (Harish and Swapna, 2024).

Semantic analysis also plays a critical but complementary role in detection research. As attackers increasingly abuse scripting languages and command-line interfaces, the semantic structure of executed commands provides valuable insights into intent. However, extensive obfuscation and encoding techniques limit the effectiveness of semantic-only approaches. Empirical studies therefore advocate for detection models that integrate semantic understanding with behavioural and contextual analysis to improve robustness against evasion (Meher et al., 2025).

Operational considerations further shape detection research priorities. Enterprise-scale detection systems must operate under real-time constraints, process high volumes of telemetry data, and maintain low false positive rates. Research indicates that analytically sophisticated approaches that cannot meet these operational requirements may be impractical for deployment, regardless of their theoretical effectiveness.

In summary, the implications of fileless and LOTL attacks for detection research point toward the necessity of integrated, context-aware, and behaviour-driven detection strategies. Future research must explore detection frameworks that can correlate semantic, behavioural, and temporal signals efficiently and in real time, enabling reliable identification of stealthy attacks embedded within legitimate system activity. These considerations motivate the methodological approach presented in the subsequent chapter.

2.6 Detection Techniques for Fileless and LOTL Attacks

The detection of fileless and Living-off-the-Land (LOTL) attacks has evolved significantly in response to the limitations of traditional malware detection techniques. Unlike file-based malware, fileless attacks operate primarily through legitimate system tools, scripting engines, and in-memory execution, leaving minimal static artifacts for analysis. As a result, detection approaches have progressively shifted from signature-centric methods toward behavior-driven and context-aware techniques.

This section reviews the major detection paradigms proposed in the literature for identifying fileless and LOTL attacks. Each technique is discussed in terms of its detection principles, strengths, and inherent limitations, with emphasis on its effectiveness against stealthy, obfuscated, and context-dependent attack behavior commonly observed in modern enterprise environments.

2.6.1 Event-Based Detection

Event-based detection represents one of the earliest and most widely adopted approaches for identifying malicious activity in enterprise systems. This technique relies on the analysis of discrete system events such as process creation, command execution, registry modification, or network connection events as primary indicators of potentially malicious behaviour. Event-based detection is commonly implemented using host-based telemetry sources including system logs, endpoint sensors, and operating system audit frameworks.

The primary advantage of event-based detection lies in its simplicity and low computational overhead. By evaluating individual events as they occur, such systems can operate in near real time and scale effectively in large enterprise environments. Academic studies note that event-based detection is particularly effective for identifying known malicious actions with clear and well-defined signatures, such as the execution of blacklisted binaries or explicit policy violations.

However, the effectiveness of event-based detection diminishes significantly in the context of fileless and Living-off-the-Land (LOTL) attacks. Fileless malware often executes malicious logic through legitimate system components, generating events that are syntactically valid and indistinguishable from benign administrative activity when examined in isolation. Research

consistently highlights that many fileless attacks do not produce single events that are inherently malicious, but rather sequences of low-suspicion actions whose intent emerges only when correlated across time and context (Singh and Tripathy, 2025).

Another limitation of event-based detection is its lack of contextual awareness. Traditional event-based systems typically analyze events independently, without considering execution lineage, parent-child process relationships, or long-term behavioural patterns. Empirical studies demonstrate that modern attacks exploit this limitation by fragmenting malicious activity across multiple processes and short-lived execution stages, each of which remains below detection thresholds when evaluated individually (Ongun et al., 2021a).

Event-based detection is also vulnerable to evasion through obfuscation and command encoding. Encoded or dynamically generated commands can obscure semantic indicators of malicious intent, reducing the effectiveness of rule-based event matching and keyword inspection. Research shows that attackers deliberately leverage these techniques to generate benign-looking events that bypass event-centric detection mechanisms (Meher et al., 2025)

Despite these limitations, event-based detection remains an essential foundation for enterprise security monitoring. Modern detection research does not seek to replace event-based detection, but rather to augment it with higher-level analytical layers capable of correlating events, extracting contextual relationships, and interpreting semantic intent. This evolution from isolated event analysis toward contextual and behaviour-aware detection forms the basis for advanced detection approaches discussed in subsequent sections.

2.6.2 Rule-Based and Signature-Based Approaches

Rule-based and signature-based approaches represent some of the earliest and most widely deployed malware detection techniques in enterprise security systems. These approaches rely on predefined rules, patterns, or signatures derived from known malicious artifacts, behaviours, or command structures. Detection decisions are typically made by matching observed system activity against a database of known indicators, such as file hashes, command patterns, registry keys, or network signatures (Khalid et al., 2023).

Signature-based detection is particularly effective against known threats with stable and repeatable characteristics. By leveraging exact or approximate pattern matching, these systems

can achieve high precision and low false positive rates when detecting previously identified malware variants. Academic studies acknowledge that signature-based methods remain effective for detecting commodity malware and large-scale reuse of known attack tools.

Rule-based detection extends signature matching by incorporating heuristic logic, such as conditional statements and thresholds, to identify suspicious activity. For example, rules may flag abnormal command usage, unauthorized access attempts, or deviations from predefined security policies. These approaches offer greater flexibility than pure signature matching and can encode expert knowledge about known attack behaviours.

Despite their effectiveness in traditional threat scenarios, rule-based and signature-based approaches exhibit significant limitations in the context of fileless and Living-off-the-Land (LOTL) attacks. Because fileless malware does not rely on persistent binaries, many signature-based mechanisms fail outright due to the absence of detectable file artifacts. Research consistently demonstrates that attackers deliberately exploit this limitation by executing malicious logic in memory or through legitimate system tools, thereby bypassing file-centric detection entirely (Leng et al., 2024).

Rule-based systems also struggle with the dynamic and adaptive nature of fileless attacks. Minor variations in command syntax, encoding schemes, or execution order can easily evade static rules without altering the underlying malicious intent. Empirical studies highlight that attackers frequently employ obfuscation, command encoding, and staged execution to defeat heuristic rules while maintaining operational effectiveness.

Another fundamental challenge lies in the contextual blindness of rule-based detection. Most rules operate on isolated events or predefined conditions, lacking the ability to reason over long-term behaviour, execution sequences, or process relationships. As a result, sophisticated attacks that distribute malicious activity across multiple low-suspicion actions can evade detection by remaining below rule thresholds at each individual step (Ongun et al., 2021a).

Operational maintenance further limits the scalability of rule-based and signature-based systems. These approaches require continuous manual updates to accommodate emerging threats, evolving attacker tactics, and environment-specific behaviours. Academic literature notes that the increasing pace of adversarial adaptation makes it difficult for static rules and signatures to remain effective without incurring high operational overhead and alert fatigue.

In summary, while rule-based and signature-based approaches remain valuable for detecting known and well-characterized threats, they are fundamentally inadequate for identifying modern fileless and LOTL attacks. Their reliance on static indicators, limited contextual reasoning, and susceptibility to evasion underscore the need for more adaptive and behaviour-aware detection strategies discussed in subsequent sections.

2.6.3 Behavioral and Log-Correlation Detection

Behavioural and log-correlation detection approaches extend event-based and rule-based techniques by analyzing sequences of events and their relationships over time. Rather than evaluating individual events in isolation, these approaches aim to identify malicious activity by correlating multiple system logs, execution traces, and behavioural signals. Academic literature recognizes behavioural detection as a critical advancement for addressing stealthy attacks that evade static signatures, particularly in the context of fileless and Living-off-the-Land (LOTL) threats (Khalid et al., 2023).

Behavioural detection models typically focus on identifying deviations from expected system behaviour. This may include abnormal process execution patterns, unusual parent-child relationships, excessive use of administrative tools, or atypical execution durations. Log-correlation techniques aggregate signals from multiple telemetry sources such as process logs, authentication records, and command execution traces to construct a higher-level view of system activity. Studies demonstrate that correlating such signals improves detection accuracy compared to isolated event analysis.

In the context of fileless and LOTL attacks, behavioural and correlation-based detection offers clear advantages. Because malicious actions often manifest as sequences of low-suspicion events, correlating execution order, timing, and process lineage can reveal attack patterns that are otherwise invisible at the event level. Empirical research highlights that techniques such as process tree analysis and temporal correlation are particularly effective for identifying chained LOTL executions and abnormal propagation behaviour (Ongun et al., 2021a).

Despite these strengths, behavioural and log-correlation approaches also exhibit important limitations. Many correlation systems rely on manually defined heuristics, thresholds, or expert-defined patterns to link events and infer malicious behavior. Such approaches can be brittle, environment-specific, and difficult to maintain as attacker tactics evolve. Studies report

that static correlation rules may fail to generalize across diverse enterprise environments or adapt to novel attack variants.

Another limitation concerns scalability and operational complexity. As the number of correlated signals and telemetry sources increases, correlation engines may experience performance bottlenecks or generate excessive alerts. Research indicates that balancing detection sensitivity with operational feasibility remains a significant challenge, particularly in high-throughput enterprise environments (Roy and Chen, 2024).

Furthermore, while behavioural and log-correlation approaches capture *how* actions unfold over time, they often lack deep semantic understanding of executed commands or scripts. As a result, benign and malicious behaviours that share similar structural patterns may be difficult to distinguish without incorporating semantic analysis or learned representations.

In summary, behavioural and log-correlation detection represents a significant improvement over event-based and rule-based approaches by introducing temporal and contextual reasoning. However, their reliance on manual heuristics, limited adaptability, and lack of semantic depth constrain their effectiveness against modern fileless and LOTL attacks. These limitations motivate the adoption of learning-based and hybrid detection approaches discussed in subsequent sections.

2.6.4 Machine Learning-Based Detection

Machine learning-based detection approaches have been increasingly adopted to address the limitations of rule-based and heuristic-driven security systems. Unlike static detection mechanisms, machine learning models learn patterns from data and can generalize beyond explicitly defined rules. Academic literature highlights that machine learning enables detection systems to adapt to evolving attack behaviours, making it particularly attractive for detecting sophisticated threats such as fileless and Living-off-the-Land (LOTL) attacks (Khalid et al., 2023).

Machine learning techniques applied to malware detection can be broadly categorized into supervised, unsupervised, and semi-supervised approaches. Supervised models are trained on labelled datasets to distinguish between benign and malicious behaviour, while unsupervised and anomaly-based models aim to identify deviations from normal system activity without

requiring explicit labels. Studies demonstrate that supervised learning often achieves higher detection accuracy when high-quality labelled data is available, whereas unsupervised methods are useful for discovering previously unseen attack patterns.

In fileless and LOTL detection, machine learning models typically operate on features extracted from system logs, process behaviour, command execution traces, or network activity. These features may include statistical summaries, behavioural indicators, or encoded representations of commands and scripts. Research shows that machine learning can effectively capture subtle patterns across multiple features that are difficult to express through manual rules or heuristics (Harish and Swapna, 2024).

Despite their advantages, machine learning based detection approaches also face significant challenges. One major limitation is their dependence on feature engineering and data quality. Poorly selected features or biased datasets can lead to overfitting, reduced generalization, or blind spots when attackers change tactics. Academic studies emphasize that the effectiveness of machine learning models is strongly influenced by how behavioural and contextual information is represented in the input data (Chidambaram and M P, 2025).

Another challenge lies in model interpretability and operational trust. Many machine learning models, particularly complex or deep architectures, function as black boxes, making it difficult for analysts to understand or validate detection decisions. This lack of transparency can hinder adoption in enterprise environments where explainability and analyst trust are critical. Research increasingly highlights the importance of explainable machine learning techniques to support operational decision-making (Elongha et al., 2025).

Scalability and real-time constraints further complicate the deployment of machine learning based detection systems. Training and inference costs, data volume, and latency requirements must be carefully balanced to ensure practical usability in high-throughput environments. Studies indicate that models optimized for offline analysis may struggle to meet the performance demands of real-time endpoint or log-based monitoring systems.

In summary, machine learning-based detection represents a significant advancement over traditional detection approaches by enabling adaptive, data-driven analysis of complex attack behaviours. However, limitations related to feature representation, interpretability, and operational constraints suggest that machine learning alone is insufficient for robust detection

of fileless and LOTL attacks. These observations motivate hybrid detection strategies that integrate machine learning with behavioural context and semantic understanding, as discussed in subsequent sections.

2.6.5 Natural Language Processing (NLP) and Semantic-based Detection

Natural Language Processing (NLP) and semantic-based detection approaches have emerged as an important direction in malware detection research, particularly for addressing script-based and command-driven attacks. Unlike traditional feature engineering that focuses on statistical or structural attributes, semantic-based approaches aim to capture the *meaning* and *intent* of executed commands, scripts, and textual artifacts. Academic studies highlight that NLP techniques are especially relevant for detecting fileless and Living-off-the-Land (LOTL) attacks, where malicious behavior is often embedded within command-line executions and scripting languages (Khalid et al., 2023)

Semantic-based detection typically involves transforming commands, scripts, or log messages into vector representations using NLP models such as word embeddings, sequence encoders, or transformer-based architectures. These representations enable detection models to identify similarities in intent across syntactically different commands, thereby improving robustness against minor variations, obfuscation, or attacker customization. Research demonstrates that semantic embeddings can effectively capture high-level behavioural intent that is difficult to express using handcrafted rules or shallow features (Elongha et al., 2025).

In the context of fileless attacks, NLP-based approaches have been widely applied to PowerShell commands, shell scripts, and command-line arguments. Studies show that semantic analysis can reveal suspicious intent even when commands are encoded, reordered, or partially obfuscated, making it a valuable complement to traditional detection mechanisms. As a result, semantic-based models often outperform rule-based approaches when dealing with polymorphic and evasive script-based attacks (Meher et al., 2025).

Despite these advantages, semantic-based detection approaches also exhibit important limitations. One key challenge is the lack of contextual grounding. Semantic models typically analyze commands or scripts as independent textual units, without incorporating execution context such as process lineage, execution timing, or system interaction patterns. Empirical

research highlights that benign administrative commands and malicious commands may share similar semantic structures, leading to ambiguity when context is ignored.

Another limitation relates to obfuscation strategies that distort semantic signals. While NLP models can tolerate certain levels of syntactic variation, aggressive encoding, runtime deobfuscation, or dynamically generated commands may reduce semantic clarity. Studies indicate that semantic-only detection approaches may fail when malicious intent is fragmented across multiple commands or execution stages, each appearing benign in isolation.

Operational considerations further constrain semantic-based detection. NLP models particularly deep or transformer-based architectures can introduce significant computational overhead during inference. Research shows that deploying such models in real-time, high-throughput enterprise environments requires careful optimization to balance detection accuracy with latency and resource constraints (Santoso et al., 2024).

In summary, NLP and semantic-based detection approaches provide powerful capabilities for understanding command-level intent and addressing evasion through syntactic variation. However, their lack of execution context, sensitivity to advanced obfuscation, and operational overhead limit their effectiveness as standalone solutions. These observations motivate the integration of semantic analysis with contextual and behavioural modelling, forming the basis for hybrid detection strategies discussed in subsequent sections.

2.6.6 Statistical and Anomaly Detection Techniques

Statistical and anomaly detection techniques aim to identify malicious activity by modelling normal system behaviour and flagging deviations from expected patterns. Unlike signature-based or supervised learning approaches, anomaly detection does not require prior knowledge of specific attack signatures, making it appealing for detecting novel or previously unseen threats. Academic literature highlights that such techniques are particularly relevant in environments where labelled attack data is scarce or rapidly evolving, including fileless and Living-off-the-Land (LOTL) attack scenarios (Khalid et al., 2023).

Statistical anomaly detection methods typically rely on mathematical models that characterize normal behaviour using metrics such as frequency distributions, execution durations, event counts, or resource utilization patterns. Deviations beyond predefined thresholds are treated as

potential anomalies. Studies show that these approaches can effectively identify abnormal system states, such as unusual process execution rates or atypical command frequencies, which may indicate malicious activity.

More advanced anomaly detection approaches employ machine learning techniques such as clustering, density estimation, or probabilistic modelling to learn normal behavioural patterns from historical data. These models can adapt to complex and high-dimensional data distributions, enabling the detection of subtle deviations that are difficult to capture using simple statistical rules. Research demonstrates that unsupervised anomaly detection can uncover stealthy attack behaviours that evade signature-based systems (Chandola et al., 2009).

In the context of fileless and LOTL attacks, anomaly detection offers clear advantages but also faces significant limitations. While fileless attacks may deviate from normal execution patterns, many legitimate administrative activities such as system maintenance or incident response also produce anomalous behaviour. Empirical studies highlight that anomaly detection systems often suffer from high false positive rates in enterprise environments, where behavioural variability is substantial (Harish and Swapna, 2024).

Another limitation concerns the lack of semantic understanding in anomaly-based approaches. Statistical deviations alone do not convey intent, making it difficult to distinguish between benign anomalies and malicious behaviour. For example, an unusual PowerShell execution may be benign during system updates but malicious in other contexts. Research consistently shows that anomaly detection models without contextual or semantic grounding struggle to provide actionable and interpretable alerts.

Operational challenges further constrain the effectiveness of anomaly detection techniques. These models often require continuous retraining to adapt to evolving baseline behavior, and abrupt changes in system usage can lead to model drift. Studies indicate that maintaining stable anomaly detection performance in dynamic enterprise environments is non-trivial and may incur significant operational overhead (Gama et al., 2014).

In summary, statistical and anomaly detection techniques provide valuable capabilities for identifying deviations from normal behavior and detecting unknown threats. However, their susceptibility to false positives, lack of semantic and contextual understanding, and operational complexity limit their effectiveness as standalone solutions for detecting fileless and LOTL

attacks. These limitations motivate the integration of anomaly detection with contextual, semantic, and learning-based approaches in hybrid detection frameworks.

2.6.7 Hybrid and Multi-Stage Detection Systems

Hybrid and multi-modal detection approaches have emerged as a response to the limitations of single-technique malware detection systems. Rather than relying on a single analytical paradigm such as signatures, behavioural heuristics, machine learning, or semantic analysis hybrid approaches integrate multiple detection techniques to leverage their complementary strengths. Academic literature increasingly recognizes hybrid detection as a promising direction for addressing sophisticated threats such as fileless and Living-off-the-Land (LOTL) attacks (Khalid et al., 2023).

Hybrid detection systems typically combine multiple data sources and analytical perspectives, including event-level telemetry, behavioural features, statistical patterns, and semantic representations. For example, event-based detection may provide real-time visibility, while behavioural correlation captures execution context, and machine learning models learn complex patterns across features. Studies demonstrate that integrating these heterogeneous signals improves detection robustness and reduces reliance on any single fragile indicator (Leng et al., 2024).

Multi-modal approaches are particularly effective in the context of fileless and LOTL attacks, where malicious behaviour is deliberately fragmented across multiple execution stages and system components. Research shows that combining contextual information such as process lineage and execution timing with semantic analysis of commands or scripts enables detection systems to uncover malicious intent that is not apparent from isolated events or features alone (Ongun et al., 2021a).

Several recent studies propose hybrid frameworks that fuse behavioral features with learned representations derived from machine learning or deep learning models. These frameworks often employ ensemble techniques, feature fusion, or layered detection pipelines to aggregate signals from different analytical components. Empirical evaluations consistently report improved detection accuracy and resilience against evasion compared to single-model approaches (Elongha et al., 2025).

Despite their advantages, hybrid and multi-modal detection approaches introduce new challenges. System complexity increases as multiple models and data pipelines must be integrated and maintained. Additionally, balancing detection accuracy with computational efficiency remains a key concern, particularly for real-time deployment in enterprise environments. Research highlights that effective hybrid systems require careful design to ensure scalability, interpretability, and operational feasibility (Roy and Chen, 2024).

In summary, hybrid and multi-modal detection approaches represent a natural evolution of malware detection research, addressing the shortcomings of individual techniques through integration and fusion. By correlating semantic, behavioral, and contextual signals, these approaches offer a more resilient defense against fileless and LOTL attacks. This trend provides a strong foundation for the detection strategies discussed in subsequent sections, including algorithm selection and comparative analysis.

2.6.8 Machine Learning Algorithm Comparison and Selection Rationale

The selection of detection algorithms for fileless and Living-off-the-Land (LOTL) attacks must account for both analytical effectiveness and practical deployment constraints. As discussed in previous sections, modern malware detection requires the integration of behavioral, contextual, and semantic signals. Consequently, algorithm design choices are driven not only by detection accuracy, but also by scalability, interpretability, and real-time feasibility within enterprise environments (Khalid et al., 2023).

One key consideration in algorithm selection is the representation of execution context. Graph-based representations are increasingly adopted to model relationships between processes, commands, and system events, as they naturally capture parent-child relationships, execution flows, and interaction patterns. However, academic studies note that not all graph-based detection approaches require full graph neural network (GNN) architectures. In many real-time scenarios, explicit graph feature extraction such as node degree, execution duration, and temporal statistics can provide sufficient contextual information with significantly lower computational overhead (Harish and Swapna, 2024).

Another important consideration is the balance between model complexity and operational constraints. Deep learning models, including GNNs and transformer-based architectures, offer strong representational power but often require substantial computational resources, extensive

training data, and careful tuning. Research highlights that such models may be impractical for real-time endpoint or log-based detection, particularly in high-throughput enterprise environments. As a result, lighter-weight models such as multilayer perceptrons (MLPs) operating on well-designed feature representations are frequently preferred for real-time detection pipelines (Elongha et al., 2025).

Feature design also plays a central role in algorithm selection. For fileless and LOTL detection, effective features often include behavioural attributes (e.g., execution duration, process relationships), contextual indicators (e.g., parent-child degree statistics), and semantic representations derived from command or script analysis. Studies emphasize that combining heterogeneous feature types through feature fusion enables simpler models to achieve competitive performance while maintaining interpretability and efficiency (Leng et al., 2024)

Interpretability is another critical design consideration, particularly in enterprise security operations where analysts must understand and trust detection outcomes. Complex models with limited explainability may hinder adoption, even if they demonstrate high detection accuracy. Academic literature increasingly advocates for the use of interpretable models or explainability techniques such as feature attribution methods to support transparent decision-making and facilitate incident investigation (Capuano et al., 2022).

Finally, algorithm selection must consider adaptability and maintainability. Detection models must remain effective as attacker tactics evolve, without requiring frequent manual rule updates or extensive retraining. Research indicates that modular designs where contextual feature extraction, semantic analysis, and classification are decoupled offer greater flexibility and resilience compared to monolithic detection architectures.

In summary, algorithm selection for fileless and LOTL detection is guided by the need to balance contextual modeling, semantic understanding, computational efficiency, and interpretability. Rather than favoring a single complex model, recent research supports the use of hybrid designs that combine explicit contextual representations with efficient learning algorithms. These considerations inform the comparative evaluation and methodological choices discussed in the following sections.

To further clarify the design considerations discussed above, Table 2.8 summarizes common machine learning-based detection approaches for fileless malware, highlighting their feature

representations, strengths, and inherent limitations. This comparison illustrates that while each approach offers distinct advantages, none of them alone sufficiently addresses the combined requirements of semantic understanding, contextual awareness, and real-time feasibility in enterprise environments.

Table 2.8 Comparison of Machine Learning Approaches for Fileless Malware Detection

Approach	Feature Representation	Strengths	Limitations
Rule-based / Signature-based	Static rules, expert-defined patterns	High interpretability, low overhead	Ineffective against fileless and obfuscated attacks
Classical ML (SVM, RF)	Hand-crafted behavioural features	Efficient, interpretable	Limited semantic understanding, feature-dependent
Deep Learning	Automatically learned representations	High detection accuracy	High latency, low explainability
Graph-based ML	Process and execution graphs	Captures contextual relationships	Computational overhead, limited semantic insight
Context–Semantic Fusion (Proposed)	Graph context + semantic command-line features	Balanced accuracy, real-time capable, explainable	Requires multi-source feature extraction

Table 2.8 summarizes the key characteristics of commonly used machine learning approaches for fileless malware detection. Traditional rule-based and classical machine learning models provide efficiency and interpretability but lack the capability to capture complex execution semantics. Deep learning approaches improve representation learning but introduce challenges related to inference latency and limited transparency. Graph-based methods effectively model contextual relationships but may not capture semantic intent embedded in command-line arguments.

Based on this comparison, the proposed context-semantic fusion approach is selected to balance detection effectiveness, real-time feasibility, and explainability. By integrating complementary feature representations, the model addresses the limitations of single-modality approaches while remaining suitable for deployment in streaming detection environments.

2.6.9 Summary of Key Weaknesses in Current Detection Techniques

The analysis of detection techniques presented in Sections 2.6.1 through 2.6.8 reveals several recurring weaknesses that limit the effectiveness of existing approaches against fileless and Living-off-the-Land (LOTL) attacks. Although each detection paradigm addresses specific aspects of malicious behaviour, no single technique adequately captures the combined characteristics of stealth, adaptability, and contextual complexity exhibited by modern attacks.

A fundamental weakness across many detection approaches is their reliance on isolated indicators. Event-based and rule-based systems typically analyze individual events or predefined patterns without sufficient contextual correlation. As discussed earlier, fileless and LOTL attacks deliberately fragment malicious behaviours across multiple low-suspicion actions, allowing them to evade detection thresholds when events are evaluated independently (Khalid et al., 2023).

Another key limitation lies in the lack of semantic understanding. While machine learning and statistical approaches can model behavioural patterns, many fail to interpret the underlying intent of executed commands or scripts. Conversely, semantic-based detection techniques often analyze commands in isolation, without incorporating execution context such as process lineage or temporal relationships. This separation leads to ambiguity, where benign administrative activity and malicious actions may appear similar when viewed through a single analytical lens (Leng et al., 2024).

Adaptability and generalization also remain significant challenges. Rule-based systems require continuous manual updates, while supervised machine learning models depend heavily on labeled data that may not reflect emerging attack variants. Anomaly detection approaches attempt to address unknown threats but frequently suffer from high false positive rates in dynamic enterprise environments. Studies indicate that these limitations reduce the long-term robustness of detection systems as attacker tactics evolve (Kara, 2023).

Operational and deployment constraints further exacerbate these weaknesses. Many advanced detection models introduce substantial computational overhead, limiting their suitability for real-time monitoring in high-throughput environments. Additionally, limited interpretability in complex models can hinder analyst trust and slow incident response. Research emphasizes that

detection effectiveness must be evaluated not only by accuracy, but also by scalability, explainability, and operational feasibility (Capuano et al., 2022).

Collectively, these weaknesses highlight a broader pattern in current detection research: approaches that focus on a single dimension whether events, rules, behaviour, semantics, or anomalies are inherently limited when confronted with modern fileless and LOTL attacks. Addressing these challenges requires detection strategies that integrate multiple analytical perspectives, enabling holistic reasoning over behavioural context, semantic intent, and temporal execution patterns.

2.6.10 Comparative Review of Existing Fileless and LOTL Detection Frameworks

Existing detection frameworks for fileless and Living-off-the-Land (LOTL) attacks can be broadly grouped into several categories based on their primary analytical paradigm: (1) graph-based behavioural/contextual modelling, (2) semantic and command-focused detection, (3) log-sequence deep learning approaches, (4) anomaly/statistical frameworks, and (5) hybrid/fusion-based systems. Comparative analysis across these categories highlights that each framework type addresses specific aspects of fileless/LOTL behaviour, yet often exhibits gaps in either semantic understanding, contextual reasoning, real-time feasibility, or explainability (Khalid et al., 2023).

Graph-based and contextual frameworks focus on capturing relationships between system entities (e.g., processes, files, network connections) to detect abnormal interaction patterns. Frameworks such as Graphite and related graph-centric approaches demonstrate strong capability in representing behavioural context and process relationships for fileless execution patterns, particularly when attackers distribute actions across multiple events. However, many graph-centric systems either depend on handcrafted graph features or require computationally intensive graph learning mechanisms, which can complicate real-time deployment and model interpretability in high-throughput environments (Wakodikar et al., 2026).

Semantic and command-focused frameworks target the textual and syntactic structure of commands and scripts commonly abused in LOTL attacks. Approaching approaches analyze command-line patterns and apply learning to reduce manual rule dependency, making them effective against variations in attacker syntax and command structure. Nevertheless, semantic-only approaches remain challenged by contextual ambiguity: benign administrative commands

may resemble malicious usage unless correlated with process lineage, timing, and surrounding behavioral signals (Harish and Swapna, 2024; Ongun et al., 2021a).

Log-sequence deep learning frameworks model system activity as temporal sequences of events, using architectures such as LSTM-based models to learn patterns in log streams. Such approaches can capture temporal dependencies and provide strong detection performance in controlled settings, but they may require substantial training data and can introduce inference overhead, affecting deployability in real-time enterprise monitoring. Additionally, sequence models may struggle to capture explicit execution relationships (e.g., parent–child process structure) unless graph/context features are incorporated into the representation (Kusuma et al., 2025; Wakodikar et al., 2026).

Anomaly and statistical frameworks detect deviations from baseline behavior using frequency analysis, statistical thresholds, or unsupervised models. Systems such as LogSHIELD demonstrate that real-time anomaly detection can be achieved efficiently and can be effective for identifying unusual event patterns at scale. However, anomaly-based frameworks frequently suffer from false positives in dynamic enterprise environments and often lack semantic understanding to explain *why* an anomaly is malicious, limiting operational trust and interpretability (Roy and Chen, 2024; Sudhakar and Kumar, 2020).

Hybrid and fusion-based frameworks aim to integrate multiple feature types and detection signals, such as behavioral/contextual attributes and semantic representations. Multi-feature fusion research indicates that combining heterogeneous signals improves robustness against fileless evasion tactics and reduces the reliance on any single detection dimension. Despite this progress, existing hybrid approaches vary in how they balance real-time feasibility, interpretability, and the explicit integration of context (process relationships) with semantic intent (command meaning) in a unified decision model (Chidambaram and M P, 2025; Leng et al., 2024)

In summary, comparative review of existing frameworks suggests a consistent trend: approaches that emphasize **context without semantics** risk missing intent-based LOTL patterns, while approaches that emphasize **semantics without context** face ambiguity and false positives. Sequence and anomaly approaches improve temporal visibility but may struggle with explicit relationship modeling and operational explainability. These comparative insights

motivate detection frameworks that explicitly combine contextual execution relationships with semantic understanding under real-time constraints an approach aligned with modern hybrid detection research directions.

Table 2.9 Detection Technique Comparison for Fileless and LOTL Attacks

No	Paper / Framework	Detection Paradigm	Feature Space / Input	Unit of Detection	Real-Time Capability	Explainability
1	Graphite (2024)	Graph-based (ML-assisted)	Process execution graph (ETW events)	Process-level execution graph	Yes (near real-time)	Not explicitly addressed
2	LogSHIELD	Graph-based (statistical anomaly detection)	Log frequency & temporal patterns	Event-level sequence	Yes (real-time)	Limited (heuristic-based)
3	LOTL Command Detection using Active Learning (2022)	Command-centric ML (Active Learning)	Command-line features & metadata	Command-level	No (offline / semi-online)	Limited (model-driven)
4	Real-Time Fileless Malware Detection Using LSTM (2023)	Sequence-based Deep Learning	System log sequences	Log-level temporal sequence	Yes (near real-time)	Not supported

No	Paper / Framework	Detection Paradigm	Feature Space / Input	Unit of Detection	Real-Time Capability	Explainability
5	Hybrid Deep Learning with CNN & Autoencoder (2023)	Hybrid Deep Learning (sequence & anomaly)	Log features & reconstructed error patterns	Log-level temporal sequence	Near real-time	Not supported
6	Multi-Feature Fusion of Complex Attack Vectors (2022)	Feature fusion (ML-based)	Behavioral + statistical features	Event & session-level	No (offline analysis)	Limited (feature-level)
7	MemLOL Dataset-Driven Detection (2023)	Classical ML (feature-based)	Memory artifacts & LOLBin usage features	Process-level	No	Limited (feature-based)
8	Robust Detection of Evasive Fileless PowerShell Malware (2025)	ML-based behavioral detection	PowerShell command features	Command-level	No (offline)	Limited (feature attribution)
9	Endpoint Detection and Response for Fileless & LOLBins (2024)	Multi-layer detection (rule + ML)	Endpoint telemetry & behavioral indicators	Event & process-level	Yes (real-time)	Limited (analyst-driven)

No	Paper / Framework	Detection Paradigm	Feature Space / Input	Unit of Detection	Real-Time Capability	Explainability
10	Proposed Research Direction (Literature Trend)	Hybrid context-semantic detection	Contextual process graph features + semantic embeddings	Process & command-level	Yes (real-time)	Explicit (XAI-supported)

(Wakodikar et al., 2026); (Roy and Chen, 2024); (Ongun et al., 2021a); (Kusuma et al., 2025); (Elongha et al., 2025); (Leng et al., 2024); (Chidambaram and M P, 2025); (Meher et al., 2025); (Harish and Swapna, 2024); (Alan., 2026)

Table 2.8 summarizes the primary machine learning–based detection approaches commonly applied to fileless malware detection, comparing their feature representations, strengths, and inherent limitations. The table highlights fundamental trade-offs between interpretability, adaptability, and computational complexity across different algorithmic categories.

Rule-based and signature-based approaches rely on static rules and expert-defined patterns, offering high interpretability and low computational overhead. These characteristics make them suitable for real-time enforcement of known policies and detection of previously identified threats. However, their dependence on predefined signatures renders them ineffective against fileless and obfuscated attacks, where malicious behavior is executed through legitimate system components without persistent artifacts.

Classical machine learning approaches, such as Support Vector Machines (SVM) and Random Forests (RF), improve detection adaptability by leveraging handcrafted behavioral features. These models are generally efficient and more flexible than rule-based systems, while maintaining a degree of interpretability. Nevertheless, their effectiveness remains strongly dependent on feature engineering, and they typically lack deep semantic understanding of commands or scripts, limiting their robustness against sophisticated LOTL attack techniques.

Deep learning approaches automatically learn feature representations from data, enabling them to capture complex and non-linear attack patterns. As a result, deep learning models often achieve high detection accuracy in controlled or offline settings. However, these models

commonly introduce higher inference latency and reduced explainability, which can pose challenges for real-time deployment and analyst trust in enterprise environments.

Overall, the comparison in Table 2.8 indicates that no single machine learning paradigm sufficiently satisfies the combined requirements of semantic understanding, contextual awareness, real-time feasibility, and interpretability for fileless malware detection. These observations motivate the exploration of hybrid detection strategies that balance lightweight classification with carefully designed contextual and semantic feature representations.

2.7 Analytical Models and Limitations

2.7.1 Graph-Based Analytical Models

Graph-based analytical models are widely used in cybersecurity research to represent complex relationships among system entities, such as processes, files, network connections, and execution events. By modeling system activity as a graph, these approaches enable explicit representation of structural and relational information, including parent-child process relationships, execution chains, and interaction patterns. Academic literature highlights that graph-based models are particularly suitable for analyzing fileless and Living-off-the-Land (LOTL) attacks, where malicious behavior is distributed across multiple system components rather than concentrated in a single artifact (Khalid et al., 2023).

Graph-based analysis in malware detection can be broadly categorized into two classes: **explicit graph feature extraction** and **graph learning-based approaches**. Explicit graph feature extraction involves computing structural properties such as node degree, execution depth, temporal duration, and connectivity patterns from process or event graphs. These features are then used as inputs to statistical or machine learning classifiers. In contrast, graph learning approaches, including Graph Neural Networks (GNNs), aim to learn representations directly from graph structure through message passing and embedding mechanisms (Khalid et al., 2023).

Explicit graph-based analytical models offer several advantages for real-time detection. Because graph construction and feature computation can be performed incrementally as events arrive, these models are well suited for streaming telemetry and online analysis. Studies demonstrate that simple graph-derived features can capture meaningful execution context while

maintaining low computational overhead, making them practical for deployment in enterprise environments (Roy and Chen, 2024).

However, graph-based analytical models also exhibit important limitations. One key challenge lies in feature expressiveness. Explicit graph features rely on predefined structural metrics, which may not fully capture higher-order dependencies or subtle attack semantics. As a result, such models may miss complex behavioural patterns unless carefully engineered and combined with complementary feature types (Harish and Swapna, 2024).

Graph learning based approaches, such as GNNs, address some expressiveness limitations by learning node and graph embeddings automatically. Nevertheless, academic research highlights several constraints associated with GNN-based detection, including high computational cost, sensitivity to graph size, and challenges in real-time inference. These limitations can restrict their applicability in high-throughput or latency-sensitive detection pipelines (Khalid et al., 2023)

Another limitation common to both graph-based paradigms is the lack of semantic understanding. While graphs effectively model *how* entities interact, they do not inherently capture the *meaning* or intent of executed commands or scripts. Without integration with semantic analysis, graph-based models may struggle to distinguish between benign administrative behavior and malicious LOTL activity that exhibits similar structural patterns (Wakodikar et al., 2026).

In summary, graph-based analytical models provide a powerful mechanism for representing contextual relationships and execution structure in fileless malware detection. However, their limitations in semantic expressiveness, scalability, and real-time feasibility motivate their integration with complementary analytical models. These observations underscore the role of graph-based analysis as a contextual foundation rather than a standalone detection solution, aligning with hybrid detection strategies discussed in subsequent sections.

Prior studies have demonstrated that temporal and structural properties of process execution, such as execution duration and parent-child relationships, are effective indicators for identifying stealthy malware behavior. These findings provide the theoretical basis for the contextual feature design adopted in this research.

2.7.2 Semantic-Based Analytical Models

Semantic-based analytical models focus on understanding the *meaning* and *intent* of executed commands, scripts, or textual artifacts generated during system activity. Unlike graph-based or purely behavioral models that emphasize structural relationships and execution patterns, semantic-based approaches aim to capture higher-level intent by analyzing the content of commands and scripts using techniques derived from Natural Language Processing (NLP). Academic literature highlights the growing relevance of semantic analysis for detecting fileless and Living-off-the-Land (LOTL) attacks, where malicious logic is often embedded within legitimate scripting environments and command-line interfaces (Khalid et al., 2023).

Semantic-based detection models typically transform commands or scripts into numerical representations using embedding techniques. These representations may be learned through traditional NLP methods (e.g., tokenization and n-grams) or more advanced sequence encoders and transformer-based models. By mapping syntactically different but semantically similar commands into nearby regions of an embedding space, semantic models can generalize beyond exact pattern matching and reduce sensitivity to superficial obfuscation or command reordering (Ongun et al., 2021b).

In fileless attack scenarios, semantic analysis has been widely applied to PowerShell commands, shell scripts, and administrative tool usage. Research demonstrates that semantic-based models can identify malicious intent even when attackers employ encoding, aliasing, or minor syntactic variations to evade signature-based detection. As a result, semantic models often outperform rule-based approaches in detecting polymorphic LOTL command execution (Meher et al., 2025).

Despite these strengths, semantic-based analytical models also exhibit notable limitations. A primary challenge is the absence of execution context. Semantic models generally analyze commands as independent textual units, without incorporating information about process lineage, execution timing, or interaction with other system components. This limitation can lead to ambiguity, as benign administrative commands may appear semantically similar to malicious commands when context is not considered (Harish and Swapna, 2024).

Another limitation arises from advanced obfuscation techniques. While semantic embeddings can tolerate certain levels of syntactic variation, aggressive encoding schemes, runtime

deobfuscation, or dynamically generated commands may obscure meaningful semantic signals. Studies indicate that semantic-only detection approaches may fail when malicious intent is distributed across multiple execution stages, each appearing benign in isolation (Gama et al., 2014).

Operational constraints further affect the applicability of semantic-based models. Deep semantic encoders, particularly transformer-based architectures, can introduce significant computational overhead during inference. Research highlights that deploying such models in real-time enterprise environments requires careful trade-offs between model complexity, latency, and detection accuracy (Wu et al., 2022).

In summary, semantic-based analytical models provide powerful capabilities for capturing command-level intent and addressing evasion through syntactic variation. However, their lack of contextual grounding, sensitivity to advanced obfuscation, and computational demands limit their effectiveness as standalone detection solutions. These limitations motivate the integration of semantic analysis with contextual and behavioural models, forming a key component of hybrid detection architectures discussed in subsequent sections.

Recent advances in semantic analysis of command-line activity show that latent execution intent can be captured through embedding-based representations, enabling robustness against obfuscation and script-based attacks. This observation underpins the semantic analysis component proposed in this study.

2.7.3 Hybrid Behavioral Models

Hybrid behavioural models combine multiple analytical perspectives to detect malicious activity by jointly considering execution behaviour, contextual relationships, and semantic intent. Unlike single-paradigm approaches that focus exclusively on structure, content, or statistical deviation, hybrid models integrate heterogeneous features to provide a more comprehensive view of system activity. Academic literature increasingly recognizes hybrid behavioural modelling as a promising strategy for detecting fileless and Living-off-the-Land (LOTL) attacks, which deliberately distribute malicious behaviour across multiple dimensions to evade isolated detection mechanisms (Khalid et al., 2023)

Hybrid behavioural approaches typically fuse contextual features such as process lineage, execution duration, and interaction patterns with semantic representations derived from command or script analysis. These features may be combined at different stages of the detection pipeline, including early feature fusion, late decision fusion, or layered detection architectures. Studies demonstrate that integrating context and semantics improves robustness against obfuscation, reduces ambiguity between benign and malicious behaviour, and enhances generalization across diverse attack scenarios.

In fileless and LOTL attack scenarios, hybrid behavioural models address key limitations observed in graph-only and semantic-only approaches. Contextual modelling captures *how* actions unfold within the system, while semantic analysis captures *what* actions are intended to achieve. Empirical research highlights that malicious intent often emerges only when these dimensions are analyzed jointly, particularly for attacks that rely on legitimate administrative tools and multi-stage execution chains (Harish and Swapna, 2024).

Despite their advantages, hybrid behavioural models introduce new challenges. Designing effective feature fusion strategies requires careful consideration to avoid redundancy, noise amplification, or excessive model complexity. Additionally, hybrid systems may increase computational overhead if not designed with real-time constraints in mind. Research emphasizes that successful hybrid models must balance detection performance with scalability, interpretability, and operational feasibility (Roy and Chen, 2024).

Another important consideration is explainability. While hybrid models can improve detection accuracy, their integration of multiple feature types and learning components may complicate interpretation of detection outcomes. Recent studies advocate for incorporating explainability mechanisms such as feature attribution or attention visualization to support analyst understanding and trust in hybrid detection systems (Capuano et al., 2022).

In summary, hybrid behavioural models represent a natural evolution in malware detection research, addressing the fragmented limitations of single-paradigm approaches. By jointly modelling behavioural context and semantic intent, these models provide a more holistic basis for detecting fileless and LOTL attacks. However, their effectiveness depends on careful architectural design, efficient feature fusion, and the incorporation of explainability to support

real-world deployment considerations that inform the methodological choices discussed in subsequent chapters.

2.7.4 Statistical and Anomaly-Based Analytical Models

Statistical and anomaly-based analytical models aim to detect malicious activity by identifying deviations from established patterns of normal system behavior. Rather than relying on explicit signatures or labeled attack data, these models characterize baseline behavior using statistical distributions, frequency analysis, or unsupervised learning techniques. Academic literature highlights anomaly-based analysis as a valuable approach for detecting novel and previously unseen threats, including fileless and Living-off-the-Land (LOTL) attacks (Khalid et al., 2023)

Statistical anomaly models typically operate by defining normal behavior through metrics such as event frequency, execution duration, resource usage, or temporal occurrence patterns. Activities that exceed predefined thresholds or fall outside learned distributions are flagged as anomalies. These models are computationally efficient and well-suited for streaming telemetry, making them attractive for real-time monitoring in large-scale enterprise environments.

More advanced anomaly-based analytical models employ unsupervised or semi-supervised learning techniques, such as clustering, density estimation, or reconstruction-based methods. These approaches can model complex behavioral patterns without requiring labeled malicious samples. Research demonstrates that anomaly-based models can detect stealthy attack behaviors that evade signature-based systems by exploiting rare or unusual execution patterns (Chandola et al., 2009).

Despite their advantages, statistical and anomaly-based models face fundamental limitations in the context of fileless and LOTL attacks. A primary challenge is behavioral overlap between benign and malicious activity. Legitimate administrative tasks such as system maintenance, incident response, or software deployment can generate anomalous behavior similar to that of fileless attacks. Empirical studies report that this overlap often results in high false positive rates, reducing operational trust and increasing analyst workload (Harish and Swapna, 2024).

Another limitation lies in the lack of semantic and contextual understanding. Anomaly-based models identify that something is unusual, but not why it is unusual. Without insight into command intent, execution relationships, or attack objectives, anomaly alerts may be difficult

to interpret or prioritize. Research consistently emphasizes that anomaly detection alone is insufficient for actionable detection of sophisticated fileless threats (Meher et al., 2025).

Model stability and adaptability also present challenges. Statistical baselines may drift over time as system usage patterns evolve, requiring continuous retraining or recalibration. Sudden changes in workload or environment configuration can further degrade model performance. Studies highlight that maintaining reliable anomaly detection in dynamic enterprise systems is non-trivial and demands careful operational management (Gama et al., 2014).

In summary, statistical and anomaly-based analytical models provide efficient mechanisms for identifying deviations from normal behavior and detecting unknown threats. However, their susceptibility to false positives, limited interpretability, and lack of semantic grounding constrain their effectiveness as standalone solutions for fileless and LOTL detection. These limitations reinforce the need to integrate anomaly-based analysis with contextual and semantic models within hybrid detection architectures.

2.7.5 Explainability and Analyst Trust

Explainability has become a critical requirement in modern malware detection systems, particularly in enterprise environments where security analysts must interpret, validate, and respond to detection outcomes. While advanced machine learning and deep learning models offer strong detection capabilities, their adoption is often hindered by limited transparency in decision-making processes. Academic literature emphasizes that detection accuracy alone is insufficient; analysts must also understand *why* a system flags an activity as malicious to build trust and support effective incident response (Capuano et al., 2022).

In the context of fileless and Living-off-the-Land (LOTL) attacks, explainability plays an especially important role. These attacks frequently exploit legitimate system tools and administrative actions, making false positives costly and disruptive. Without clear explanations, analysts may struggle to distinguish between benign administrative behaviour and genuine malicious activity. Studies indicate that opaque detection models can increase alert fatigue and reduce analyst confidence, ultimately undermining operational effectiveness (Khalid et al., 2023).

Explainable Artificial Intelligence (XAI) techniques aim to address these challenges by providing interpretable insights into model decisions. Feature attribution methods such as SHAP (SHapley Additive exPlanations) and LIME quantify the contribution of individual features to a model's output, enabling analysts to understand which behavioural or semantic factors influenced a detection decision. Research shows that such techniques improve transparency, facilitate model validation, and support more informed security investigations (Lundberg and Lee, 2017).

In hybrid detection architectures, explainability also serves as a bridge between multiple analytical components. When contextual features (e.g., execution duration, process relationships) and semantic features (e.g., command embeddings) are combined, XAI methods can reveal how these heterogeneous signals interact to produce a detection outcome. This capability is particularly valuable for hybrid models, where interpretability might otherwise degrade due to increased model complexity (Tjoa and Guan, 2021).

Despite recent advances, explainability itself has limitations. Feature attribution methods depend on the underlying feature representation and model architecture; misleading or unstable explanations may arise if features are poorly defined or highly correlated. Additionally, explainability techniques introduce computational overhead, which must be considered in real-time detection scenarios. Academic studies caution that explainability should be treated as a complementary tool rather than a complete solution to trust and validation challenges (Doshi-Velez and Kim, 2017).

In summary, explainability is a foundational requirement for effective deployment of malware detection systems in enterprise environments. By enabling analysts to interpret model decisions and assess detection rationale, explainable models foster trust, reduce false positive impact, and support operational adoption. These considerations reinforce the importance of incorporating explainability mechanisms such as feature attribution into hybrid detection frameworks for fileless and LOTL attacks.

Explainability techniques such as SHAP have been increasingly adopted in security analytics to support analyst trust, model validation, and operational decision-making, particularly in high-stakes domains such as malware detection.

2.8 Evaluation Metrics for Fileless Malware Detection

Evaluation metrics play a critical role in assessing the effectiveness and practical usability of fileless malware detection systems. Unlike traditional file-based malware detection, fileless and Living-off-the-Land (LOTL) attacks exhibit high stealth, low execution footprint, and strong overlap with benign administrative behaviour. As a result, conventional metrics such as overall accuracy may provide misleading assessments of detection performance. Academic literature emphasizes the need for evaluation metrics that account for class imbalance, false positive impact, and real-time operational constraints (Khalid et al., 2023).

Evaluation metrics are essential for assessing the effectiveness and operational feasibility of malware detection systems. Traditional metrics such as accuracy, precision, recall, F1-score, and ROC-AUC are widely used in intrusion detection and machine learning research (Sokolova & Lapalme, 2009; Fawcett, 2006). However, for fileless and Living-off-the-Land (LOTL) attacks, which are characterized by class imbalance and behavioral stealth, careful metric selection is required.

Accuracy, while intuitive, may produce misleading results in highly imbalanced security datasets where benign events dominate malicious ones (Powers, 2011). Therefore, modern research emphasizes precision, recall, and F1-score to better capture detection sensitivity and false positive control (Khalid et al., 2023).

Additionally, ROC-AUC provides threshold-independent evaluation of classification performance (Fawcett, 2006), which is important when decision thresholds are tuned for operational deployment.

Beyond classification effectiveness, real-time detection systems must also evaluate latency-related metrics. Detector-side processing latency and end-to-end detection latency are increasingly recognized as critical evaluation dimensions in streaming security systems (Roy and Chen, 2024).

Evaluating the performance of fileless malware detection systems requires careful selection of evaluation metrics that reflect both detection effectiveness and operational feasibility. Unlike traditional file-based malware detection, fileless and Living-off-the-Land (LOTL) attacks are characterized by highly imbalanced datasets, stealthy execution patterns, and strong overlap with legitimate administrative behavior. As a result, commonly used metrics such as accuracy

may provide misleading assessments, while metrics that account for false positives, detection sensitivity, and threshold behavior are more appropriate for real-world deployment. Recent literature reveals variation in metric emphasis across detection paradigms. Classical anomaly detection systems often prioritize recall to minimize missed attacks, while enterprise deployment studies emphasize precision to reduce alert fatigue. Graph-based and streaming detection systems further introduce latency metrics to evaluate operational responsiveness. Table 2.10 compares commonly used evaluation metrics, highlighting their theoretical properties and suitability for fileless and LOTL detection contexts.

Table 2.10 Comparison of Evaluation Metrics Used in Malware Detection

Metric	Description	Strengths	Limitations	Used in This Study
Accuracy	Ratio of correctly classified events	Simple and intuitive	Misleading for imbalanced datasets	No 
Precision	Proportion of true positives among predicted positives	Reflects false-positive control	Sensitive to threshold selection	Yes
Recall	Proportion of true positives among actual positives	Measures detection capability	May increase false positives	Yes
F1-score	Harmonic mean of precision and recall	Balances precision and recall	Does not capture threshold behavior	Yes
ROC-AUC	Discrimination capability across thresholds	Threshold-independent evaluation	Less intuitive for operations	Yes
Detection Latency	Time required to produce detection result	Critical for real-time response	Implementation-dependent	Yes

Table 2.9 summarizes the commonly used evaluation metrics for malware detection and highlights the metrics adopted in this study. Accuracy is a general measure of classification correctness; however, it is not emphasized due to the benign-dominant nature of realistic security datasets, where high accuracy may not reflect effective detection of malicious activities.

Precision and recall are prioritized to capture complementary aspects of detection performance. Precision reflects the system's ability to minimize false positives, which is critical in real-time security monitoring to reduce analyst workload and alert fatigue. Recall measures the capability to correctly identify malicious events, ensuring that fileless and Living-off-the-Land (LOTL) attacks are not missed.

The F1-score is used to balance precision and recall, providing a single metric that reflects overall detection effectiveness under class imbalance. In addition, ROC-AUC is employed to evaluate the model's discrimination capability across different decision thresholds, offering a threshold-independent performance measure.

Beyond classification effectiveness, latency-related metrics are incorporated to assess operational feasibility. Detector-side processing latency is particularly important for real-time deployment, as timely detection directly impacts response and mitigation. Collectively, these metrics align with the objectives of this research by evaluating both detection accuracy and real-world applicability.

Based on the characteristics of fileless and LOTL attacks and the requirements of real-time deployment, this study primarily employs precision, recall, F1-score, ROC-AUC, and latency-related metrics for evaluation. These metrics collectively capture detection effectiveness, false-alarm control, threshold-independent discrimination, and operational responsiveness. Accuracy is not emphasized due to the benign-dominant nature of realistic security workloads, where high accuracy may not reflect true detection performance.

In addition to offline classification metrics, online and real-time detection systems require operational evaluation criteria. In streaming environments, performance is not only determined by classification effectiveness but also by detection timeliness. Therefore, latency-related metrics, including detector-side processing latency and end-to-end detection latency, are essential for assessing real-time applicability.

In this study, online evaluation focuses on event-level detection latency and continuous telemetry processing capability rather than traditional accuracy-based metrics. This distinction reflects the operational requirements of real-time security monitoring, where delayed detection may reduce the effectiveness of response actions. The measurement of latency under streaming conditions is further discussed in Chapter 4 to demonstrate the practical feasibility of the proposed framework.

2.9 Synthesis and Research Gap

The preceding sections have surveyed the state of the art in fileless and Living-off-the-Land (LOTL) attack detection, focusing on both the detection strategies proposed in the literature and the analytical modelling approaches adopted across recent studies. Although these works offer meaningful contributions ranging from behavioural profiling and sequence-based modelling to semantic analysis and system-level telemetry correlation the research landscape remains notably heterogeneous and, in many respects, methodologically fragmented.

More specifically, existing approaches differ substantially in the types of telemetry they rely upon, the modelling paradigms they employ, and the extent to which they capture higher-level semantic intent beyond low-level behavioural indicators. In addition, studies vary in their assumptions regarding deployment constraints, including computational overhead, data availability, and the feasibility of real-time execution in operational environments. Furthermore, despite growing attention to interpretable machine learning and explainable security analytics, explainability remains inconsistently addressed, particularly for models designed to detect multi-stage, stealth-driven attack behaviours such as fileless and LOTL campaigns.

Accordingly, this section consolidates the insights derived from the reviewed body of work, with the aim of establishing a coherent perspective on what current detection methods achieve, where they fall short, and which unresolved challenges persist. Based on this synthesis, the section then articulates the key research gaps that motivate the design and development of the proposed framework, particularly in terms of integrating multi-source telemetry, improving semantic and lifecycle-level reasoning, enabling practical real-time applicability, and supporting transparent, analyst-friendly explanations of detection outcomes.

2.9.1 Synthesis of Key Insights from the Literature

The literature reviewed in this chapter reveals a clear and consistent evolution in malware detection research, driven by the increasing prevalence of fileless and Living-off-the-Land (LOTL) attacks. Traditional detection paradigms such as signature-based and rule-based systems were initially effective against file-based threats but have become increasingly inadequate as attackers shift toward memory-resident execution, legitimate tool abuse, and stealth-oriented tactics. This evolution has necessitated a transition from static artifact inspection toward behavior-centric and context-aware detection strategies.

A key insight from the literature is the limitation of isolated event analysis. Event-based detection approaches provide essential real-time visibility but lack the contextual awareness required to identify multi-stage and low-signal attack behaviors. Rule-based extensions partially address this limitation through heuristic correlation, yet remain fragile in the face of obfuscation and minor behavioural variation. These findings underscore that fileless and LOTL attacks rarely manifest as single, overtly malicious events; instead, malicious intent emerges across sequences of seemingly benign actions.

The reviewed studies further demonstrate that behavioural and log-correlation techniques represent an important advancement by incorporating temporal and relational information. Process lineage analysis, execution timing, and cross-log correlation significantly improve detection capability for distributed attack behaviours. However, these approaches often depend on manually crafted heuristics or fixed correlation rules, limiting adaptability and scalability in dynamic enterprise environments.

Machine learning based detection approaches introduce data-driven adaptability and improved generalization across attack variants. Supervised models achieve strong performance when high-quality labelled datasets are available, while unsupervised and anomaly-based models enable detection of previously unseen threats. Nevertheless, the literature consistently highlights challenges related to feature dependency, false positive rates, concept drift, and operational feasibility particularly when models lack semantic or contextual grounding (Khalid et al., 2023).

Semantic-based analytical models provide critical insight into command-level intent, especially for script-driven and LOTL attacks. By leveraging NLP techniques and learned embeddings,

these approaches reduce reliance on exact syntax and improve resilience against superficial obfuscation. However, semantic-only detection remains insufficient due to its inability to capture execution context, leading to ambiguity between benign administrative actions and malicious behaviour.

Graph-based analytical models contribute complementary strengths by explicitly modeling execution relationships and system interactions. Process graphs and contextual features effectively represent attack structure and propagation behavior. Yet, graph-based approaches alone often lack semantic understanding and may introduce computational overhead when extended to full graph learning architectures, limiting their applicability in real-time detection scenarios.

Across the literature, a recurring theme emerges: **no single analytical paradigm adequately addresses the combined challenges of context, semantics, adaptability, real-time performance, and explainability.** This realization has driven growing interest in hybrid and multi-modal detection frameworks that integrate contextual, semantic, and behavioural features within a unified detection pipeline. Additionally, recent studies emphasize the importance of explainability and analyst trust, highlighting that operational deployment requires not only accurate detection but also transparent and interpretable decision-making.

Collectively, these insights establish a strong foundation for hybrid detection research that balances contextual awareness, semantic understanding, computational efficiency, and explainability. This synthesis informs the comparative analysis and research gaps discussed in the following sections.

2.9.2 Multi-Dimensional Comparison Matrix

This comparison focuses on architectural and methodological capabilities. Quantitative performance metrics are evaluated separately under controlled experimental settings in Chapter 4.

To consolidate the diverse detection approaches discussed throughout Chapter 2, a multi-dimensional comparison is presented in Table 2.11. This table summarizes representative academic frameworks across multiple dimensions, including telemetry scope, analytical paradigm, semantic understanding, contextual awareness, real-time capability, latency

reporting, and explainability support. Unlike performance-oriented evaluations, this comparison emphasizes architectural and methodological characteristics to highlight fundamental design trade-offs in existing fileless and Living-off-the-Land (LOTL) detection approaches.

Table 2.11 Multi-Dimensional Comparison of Fileless and LOTL Detection Frameworks

Dimension	Graphite	LogSHIELD	LOTL Cmd AL	Multi- Feature Fusion Fileless	Real- Time LSTM Logs	Proposed Framework
Primary Focus	Windows fileless	System anomalies	LOTL commands	Fileless attacks	Fileless behavior	Fileless & LOTL
Telemetry Scope	ETW	System logs	Command- line	Process & script logs	System logs	Sysmon, WMI, ETW
Dataset Type	Custom attack traces	Custom logs	Custom LOTL commands	Custom fileless traces	Log-based traces	driven behavioral dataset
Dataset Source	Lab- generated ETW traces from simulated fileless attacks	Controlled system logs collected from benign and attack scenarios	Red-team- simulated LOTL command executions	Controlled testbed executing known fileless attack vectors	Windows logs collected during benign and malicious activity	Multi-VM Windows lab executing MITRE ATT&CK fileless and LOTL techniques
ATT&CK Alignment	Partial	Partial	Yes (LOTL)	Yes	Partial	Yes (explicit)
Feature Type	Process interaction graphs	Log frequency and structural features	Command- line semantic features	Handcrafted behavioral features	Sequential log features	Contextual process features + semantic embeddings

ML / DL Model	Graph-based ML	Statistical graph analysis	Active Learning (ML)	Classical ML classifiers	LSTM (Deep Learning)	Fusion MLP (Context–Semantic)
Semantic Understanding	None	None	Strong	Limited	None	Strong
Contextual Awareness	Moderate (process graph)	Moderate	Low	Low	Low	Strong (lineage, parent, signer)
Real-Time Capability	Yes	Yes	No	Near real-time	Yes	Yes (streaming)
Latency Reporting	Partial	Partial	Not reported	Not reported	Reported (log-level)	Explicit (ms-level)
Obfuscation Handling	Weak	Weak	Medium	Medium	Medium	Strong
Explainability (XAI)	None	None	None	None	None	Yes (SHAP + attention)
Threat Coverage	Windows fileless	General anomalies	LOTL abuse	Fileless malware	Fileless malware	Windows fileless & LOTL
Primary Limitation	No semantic reasoning	No semantics	No real-time context	No explainability	No semantic awareness	Scope limited to Windows OS

Graphite (Wakodikar et al., 2026); LogSHIELD (Roy and Chen, 2024); LOTL Command Detection using Active Learning (Ongun et al., 2021b); Early Detection of Fileless Attacks Based on Multi-Feature Fusion (Leng et al., 2024); Obfuscated Fileless Memory Forensics (Carrier et al., 2022); Context-Semantic Real-Time Detection Framework (*This Work*, 2026).

Table 2.11 presents a multi-dimensional comparison of representative detection frameworks for fileless and Living-off-the-Land (LOTL) attacks in Windows environments. To ensure a fair and operationally relevant comparison, only implemented and empirically evaluated systems are included, while survey-based and taxonomy-focused studies are excluded from the main analysis.

The comparison highlights substantial differences in telemetry scope and dataset sources across prior work. Most existing approaches rely on custom or laboratory-generated datasets due to the limited availability of publicly labelled fileless and LOTL datasets. While this practice is common in the literature, the proposed framework explicitly aligns its data collection with

MITRE ATT&CK techniques to improve threat representativeness and experimental reproducibility.

To clarify methodological distinctions, the analytical paradigm is decomposed into two complementary dimensions: feature type and machine learning or deep learning (ML/DL) model. Feature types range from structural representations such as process interaction graphs and log frequency statistics to semantic representations derived from command-line or script content. In parallel, learning models span classical machine learning, graph-based analysis, and deep learning architectures such as LSTM. This separation reveals that most prior studies emphasize either contextual behaviour or semantic understanding in isolation.

As shown in the table, frameworks such as Graphite (Wakodikar et al., 2026) and LogSHIELD (Roy and Chen, 2024) provide real-time detection with moderate contextual awareness but lack semantic reasoning capabilities. Conversely, LOTL Command Active Learning (Ongun et al., 2021b) demonstrates strong semantic understanding of LOTL commands but does not support real-time operation or rich contextual analysis. Other approaches, including multi-feature fusion and LSTM-based log analysis (Leng et al., 2024), improve detection coverage but remain limited in explainability and explicit handling of LOTL abuse.

In contrast, the proposed framework uniquely integrates contextual process features and semantic embeddings within a lightweight fusion model designed for real-time streaming environments. The inclusion of explicit latency reporting and explainable AI (XAI) mechanisms further distinguishes the proposed approach from existing work, addressing critical operational requirements for analyst trust and deployment feasibility. Overall, the comparison illustrates that the proposed framework bridges key gaps in prior research by combining context awareness, semantic understanding, real-time capability, and explainability within a unified detection pipeline.

2.9.3 Discussion of Comparative Literature Findings

The comparative analysis in Table 2.11 reveals several important patterns and limitations in existing fileless and Living-off-the-Land (LOTL) detection frameworks. A primary finding is that prior approaches tend to emphasize either contextual process behaviour or semantic command understanding in isolation, rather than integrating both perspectives within a unified detection pipeline. Context-centric frameworks such as Graphite and LogSHIELD leverage

process relationships or log-level statistics to enable real-time detection, yet they lack semantic reasoning capabilities to interpret the intent of command-line or script-based activities commonly abused in LOTL attacks.

Conversely, approaches that focus on semantic analysis, such as LOTL Command Active Learning, demonstrate strong capability in identifying malicious command usage patterns but operate primarily in offline or near-offline settings. The absence of real-time execution context and process lineage information limits their effectiveness against multi-stage attacks in which benign and malicious actions are interleaved. This finding highlights a recurring trade-off in the literature between semantic expressiveness and operational deployability.

Another notable observation concerns the treatment of real-time detection and latency. While several frameworks claim real-time capability, latency reporting is often partial or entirely absent, making it difficult to assess their suitability for operational environments. Log-based deep learning approaches, such as LSTM-based detection, improve temporal modeling but still rely on delayed log aggregation and do not explicitly incorporate semantic understanding of executed commands. In contrast, few studies explicitly quantify detection latency at a granularity relevant to live incident response.

The analysis further indicates that obfuscation handling remains a challenge across most frameworks. Memory-forensics-based approaches exhibit stronger resilience to obfuscated fileless malware but incur significant computational overhead and lack streaming capability. As a result, such methods are more suitable for post-incident analysis than continuous monitoring. This reinforces the need for lightweight detection mechanisms that can operate under real-time constraints while remaining robust to evasion techniques.

From a dataset perspective, Table 2.11 shows that the majority of studies rely on custom or laboratory-generated datasets due to the scarcity of publicly available, labeled fileless and LOTL datasets. While this limitation is common across the literature, only a subset of works explicitly align attack scenarios with established threat models such as MITRE ATT&CK. The lack of standardized datasets and explicit threat alignment complicates cross-study comparison and reproducibility, underscoring a persistent research gap in this domain.

Overall, the comparative findings suggest that existing approaches address isolated aspects of fileless and LOTL detection but fall short of providing a holistic solution that combines

contextual awareness, semantic understanding, real-time capability, and explainability. These limitations motivate the need for an integrated framework capable of fusing heterogeneous feature representations and delivering interpretable detection outcomes under operational constraints, which forms the basis for the proposed approach in this research.

2.9.4 Research Gaps

Based on the multi-dimensional comparison presented in Table 2.11, several critical research gaps can be identified in existing fileless and Living-off-the-Land (LOTL) detection frameworks. Although prior studies have made significant progress in addressing specific aspects of fileless attack detection, the comparison reveals persistent limitations that hinder comprehensive, real-time, and operationally viable detection.

First, there is a lack of unified integration between contextual and semantic analysis.

Graph-based and statistical frameworks, such as Graphite and LogSHIELD, demonstrate strong capability in modelling process relationships and execution context. However, these approaches do not incorporate semantic understanding of executed commands, limiting their ability to distinguish between benign administrative behaviour and malicious LOTL activity. Conversely, semantic-focused approaches, such as LOTL Command Detection using Active Learning, effectively capture command intent but operate largely on isolated command instances, resulting in weak contextual awareness. The absence of a unified model that jointly captures *execution context* and *command semantics* remains a key gap in current research.

Second, real-time detection capability is often traded off against analytical depth.

Several frameworks achieve detailed behavioural or semantic analysis but operate in offline or near-real-time settings, with limited reporting on end-to-end detection latency. Approaches that support real-time detection frequently rely on simplified feature representations or statistical thresholds, reducing their effectiveness against multi-stage and low-signal fileless attacks. The lack of frameworks that explicitly balance semantic richness, contextual modelling, and streaming inference under real-time constraints represents a significant research gap.

Third, explainability is largely absent from existing detection frameworks.

As shown in Table 2.11, most prior approaches do not provide explicit explainability mechanisms. This limitation is particularly problematic in enterprise environments, where

analysts must interpret alerts involving legitimate administrative tools and ambiguous behavior. The absence of feature-level attribution or decision rationale undermines analyst trust and complicates incident response. There is a clear gap in integrating explainable AI techniques into fileless and LOTL detection systems without sacrificing detection performance or latency.

Fourth, dataset alignment with standardized attack frameworks remains inconsistent.

While some studies partially reference MITRE ATT&CK techniques, many datasets are custom-built without explicit alignment to standardized adversarial behaviors. This inconsistency limits reproducibility, comparability across studies, and practical relevance for enterprise deployment. The need for ATT&CK-driven behavioral datasets that capture both benign and malicious execution paths is insufficiently addressed in existing work.

Finally, existing frameworks often focus on single analytical paradigms rather than holistic detection strategies.

Most approaches prioritize either contextual modeling, semantic understanding, anomaly detection, or machine learning classification in isolation. As a result, detection capabilities remain fragmented, and systems struggle to generalize across diverse fileless and LOTL attack scenarios. The literature lacks a cohesive framework that integrates contextual process modeling, semantic representation, real-time inference, and explainability within a unified detection pipeline.

In summary, the research gaps identified through the multi-dimensional comparison highlight the need for a hybrid detection framework that jointly addresses contextual awareness, semantic understanding, real-time performance, explainability, and standardized dataset alignment. Addressing these gaps forms the basis of the proposed research methodology presented in the subsequent chapter.