Evaluating the Detection and Response Effectiveness of Wazuh Across Multiple Endpoint Threat Categories

Zakiya Tiona Williams

North Carolina A&T State University

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Department: Electrical and Computer Engineering

Major: Electrical & Computer Engineering

Research Professor: Dr. Zhijian Xie

Greensboro, North Carolina

2025

The Graduate College
North Carolina Agricultural and Technical State University

This is to certify that the Master's Thesis of

Zakiya Tiona Williams

has met the thesis requirements of
North Carolina Agricultural and Technical State University

Greensboro, North Carolina
2025

Approved by:

Dr. Zhijian Xie
Major Professor

Dr. Ahmad Patooghy
Major Professor

Dr. John C. Kelly
Committee Member

Dr. Ravinder Jain
Department Chair

Dr. Clay Gloster Jr.
Dean, The Graduate College

Biographical Sketch

Zakiya Tiona Williams is a dedicated cybersecurity researcher and graduate student pursuing a Master's degree in Electrical and Computer Engineering at North Carolina Agricultural and Technical State University. With a strong background in network security, threat detection, and incident response, Zakiya specializes in evaluating Security Information and Event Management (SIEM) solutions to enhance organizational cybersecurity. While pursuing her master's degree, Zakiya has also obtained industry recognized certifications including CCNA and CompTIA Security+, and is currently studying for CompTIA Network+, further strengthening her expertise in cloud security, SOC operations, and threat intelligence.

Before transitioning into cybersecurity, Zakiya built a distinguished career in education and technology leadership, where she spearheaded network infrastructure projects, cybersecurity training, and curriculum development. This experience has provided a unique perspective on bridging technical expertise with real world security challenges. Currently, Zakiya is conducting research on evaluating the detection and response effectiveness of Wazuh as a SIEM platform, with a focus on log ingestion performance, detection accuracy, rule diversity, and cross platform consistency across Windows and Linux endpoints. Her research aims to provide actionable insights for organizations evaluating open source security solutions.

With a relentless drive for problem solving and continuous learning, Zakiya aspires to contribute to the advancement of cybersecurity best practices and innovative threat detection methodologies.

Dedication

I dedicate this work to my parents and siblings, whose unwavering love, support, and belief in me have been my greatest source of strength. Thank you for always having my back and encouraging me to pursue my dreams.

To my family and friends, who have walked this journey with me, your support, patience, and encouragement have been invaluable. I am deeply grateful for the role each of you has played in helping me reach this milestone.

To Myreon, thank you for being my calm in the storm, for your love, encouragement, and steady support through the highs and lows of this journey. Your presence has been a source of peace, strength, and motivation when I needed it most.

This achievement is as much yours as it is mine.

Acknowledgements

I would like to express my deepest gratitude to my research advisors, Dr. Mahmoud and Dr. Patooghy, for their guidance and mentorship. Your encouragement, insightful feedback, and high expectations have pushed my thinking and learning beyond what I thought possible. Thank you for challenging me to grow as a researcher and as a professional.

I am also incredibly grateful to Dr. Kelly, Chair of the Engineering Department, for your advice, assistance, and for providing me with the resources and opportunities that have been instrumental in my academic and professional journey.

A special thank you to Dr. Horne, who took the time to teach me fundamental skills and concepts in a way that made learning accessible and impactful. You were always just a call away, ready to offer support whenever I needed it. Your patience and willingness to guide me through challenges made all the difference.

To Dr. Dogan, thank you for seeing my potential and doing everything possible to help me succeed. Our long meetings, sometimes lasting hours, even on weekends, were a testament to your dedication. Your commitment to my growth and success has meant so much, and I truly appreciate the time and effort you invested in me.

A heartfelt thank you to my colleague and mentor, Jason, for showing me the ropes, motivating me, and providing invaluable mentorship throughout my research. Your guidance has been essential in navigating both academia and preparation for the workforce.

To my friends and colleagues, Edikan Gideon, Simon Ofeanyi, Oghene Fejiroc, and Mohammed Zakaria, your encouragement and camaraderie have made this journey all the more meaningful. I truly appreciate the advice, shared knowledge, and friendships that have helped me through this process.

This work would not have been possible without each of you, and I am deeply grateful for your support.

Table of Contents

List of Figures

List of Tables

Abstract

Modern cyber defense strategies rely on accurate and timely threat detection across endpoint environments. Open source SIEM solutions have emerged as a critical resource, particularly for organizations seeking cost effective yet capable security monitoring alternatives to commercial platforms. Unlike enterprise systems, open source SIEMs lower the barrier to entry for smaller organizations and academic institutions by reducing licensing costs while still providing robust capabilities for log aggregation, threat detection, and incident response. Their flexibility and community driven development allow innovation and rapid adaptation to evolving threats. However, despite their growing adoption, there is limited empirical research examining whether these tools perform consistently across different operating systems. This gap is particularly relevant as modern enterprise networks are heterogeneous, often combining Windows and Linux endpoints. Understanding how open source SIEM platforms perform in such environments is essential for ensuring reliable threat visibility and informed security architecture decisions.

This research evaluates the detection and response effectiveness of the open source security information and event management (SIEM) platform Wazuh across Windows and Linux endpoints under controlled adversarial simulations. Using a structured attack campaign mapped to the MITRE ATT&CK framework, multiple techniques including command execution, credential access, network discovery, and persistence were simulated through CALDERA agents and custom scripts. Alert volume, rule diversity, detection latency, and severity classification were quantitatively analyzed to assess platform performance and cross operating-system detection trends.

The results indicate that Wazuh consistently detects high severity events but exhibits measurable variation in alert patterns, latency, and rule coverage between Windows and Linux endpoints. While Windows generated broader rule diversity and earlier detections, Linux produced fewer alerts concentrated at higher severity levels. These findings highlight Wazuh's strengths in critical threat visibility and its gaps in consistent cross platform coverage.

This study contributes to the growing body of research on open source security operation

center (SOC) and SIEM tools by demonstrating a rigorous, repeatable methodology for evaluating detection performance across operating systems. The results provide practical insights for security engineers, defenders, and organizations seeking to optimize endpoint monitoring strategies in resource constrained environments.

This research presents a novel architecture for assessing the performance of SIEM solutions across multiple operating systems. While most SIEM providers claim platform independence, our analysis demonstrates otherwise. The proposed framework provides a structured and repeatable approach to evaluate detection accuracy, response timeliness, and cross-platform consistency, offering valuable insights into the true operational capabilities of SIEM systems.

CHAPTER 1

Introduction

1.1 Background

In today's digital landscape, organizations face so many cyber threats that challenge the integrity, confidentiality, and availability of their information systems. The increasing sophistication of cyberattacks requires companies to have robust security measures that can detect, analyze, and respond to incidents in real time. Security Information and Event Management (SIEM) systems have emerged as pivotal tools, offering centralized solutions for monitoring and managing security events across complex IT infrastructures [12, 13].

SIEM systems integrate the functionalities of Security Information Management (SIM) and Security Event Management (SEM), enabling the collection, aggregation, and analysis of security related data from various sources within an organization's network. By correlating events from disparate systems, SIEMs provide a holistic view of an organization's security posture, facilitating the early detection of potential threats and compliance with regulatory requirements [12].

The evolution of SIEM technology has been marked by significant advancements. Modern SIEM solutions incorporate machine learning algorithms and artificial intelligence to enhance threat detection capabilities. These systems can identify anomalies and patterns indicative of malicious activity, thereby improving the accuracy and speed of incident response. Furthermore, the integration of SIEMs with cloud services has expanded their applicability, allowing organizations to monitor hybrid environments effectively [13].

Despite these advancements, traditional SIEM solutions often come with high costs and complexity, posing challenges for small to medium sized enterprises (SMEs) seeking comprehensive security measures [13, 14]. The financial and resource intensive nature of proprietary SIEMs can be prohibitive, leading organizations to explore alternative solutions that offer similar capabilities without the associated overhead.

In response to these challenges, open source SIEM platforms have gained traction, providing cost effective and customizable solutions for organizations of varying sizes. Among these, Wazuh has emerged as a prominent open source security platform that extends the capabilities of traditional SIEMs [15, 16]. Wazuh offers features such as log analysis, intrusion detection, vulnerability detection, and compliance monitoring, all within a unified framework [15].

Wazuh's architecture is designed to be scalable and flexible, supporting a wide range of operating systems and environments. Its agent based model allows for the collection of security data from endpoints, which is then analyzed and correlated to detect potential threats. Additionally, Wazuh integrates with the MITRE ATT&CK framework, enabling organizations to map detected threats to known adversary tactics and techniques, thereby enhancing threat intelligence and response strategies [15, 16].

To assess and improve the effectiveness of security measures, organizations often employ threat simulation tools that mimic real world attack scenarios. These tools enable defenders to test detection and response capabilities in controlled environments. For example, Atomic Red Team provides a library of tests mapped to the MITRE ATT&CK framework, allowing security teams to validate specific techniques. [17]. However, for this research, CALDERA was selected as the primary simulation platform due to its ability to automate multistage campaigns, emulate realistic adversary behaviors, and coordinate activity across multiple endpoints.

The integration of SIEM systems like Wazuh with threat simulation tools such as CALDERA offers a comprehensive approach to cybersecurity. By continuously testing and refining security controls, organizations can proactively address vulnerabilities and adapt to the dynamic threat landscape. This not only enhances the detection and response capabilities but also contributes to a more resilient and secure IT environment.

1.2 Motivation

The global cost of cybercrime continues to escalate at an alarming rate, with projected losses expected to reach $13.82 trillion by 2028 (see Figure 1). This rapid growth in cyber threats highlights the urgent need for robust, scalable, and affordable security solutions.

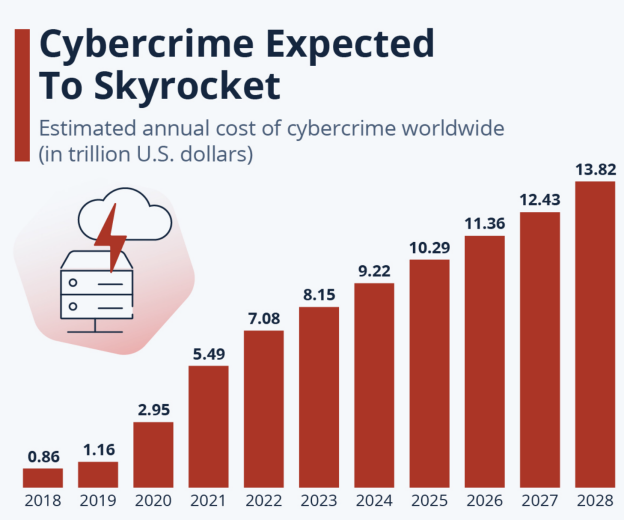


Figure 1. Projected global cost of cybercrime from 2018 to 2028, illustrating the growing economic impact of security breaches. Source: [1] .

As the cybersecurity landscape continues to evolve, small to medium sized enterprises (SMEs) face pressure to defend their digital assets against a growing number of sophisticated threats. Unlike large corporations with large budgets and dedicated security operations centers (SOCs), SMEs often operate with limited financial and technical resources. This disparity creates a significant vulnerability gap, leaving smaller organizations exposed to attacks such as ransomware, phishing, credential theft, and remote access trojans. For these enterprises, the ability to detect and respond to endpoint threats in real time is not a luxury but a necessity for maintaining operational continuity and protecting sensitive data.

Figure 2 illustrates the most common attack vectors affecting SMEs in recent years, with phishing, stolen credentials, and ransomware leading the threat landscape. These patterns highlight the importance of real time detection and correlation capabilities that a properly configured SIEM like Wazuh can deliver even for resource constrained organizations.

Security Information and Event Management (SIEM) systems are widely recognized as essential components of a mature cybersecurity strategy. These platforms offer centralized log collection, threat detection, alerting, and forensic analysis capabilities that are critical for identifying malicious behaviors across endpoints, servers, and networks [12]. However, the high licensing costs, deployment complexity, and steep learning curves associated with many

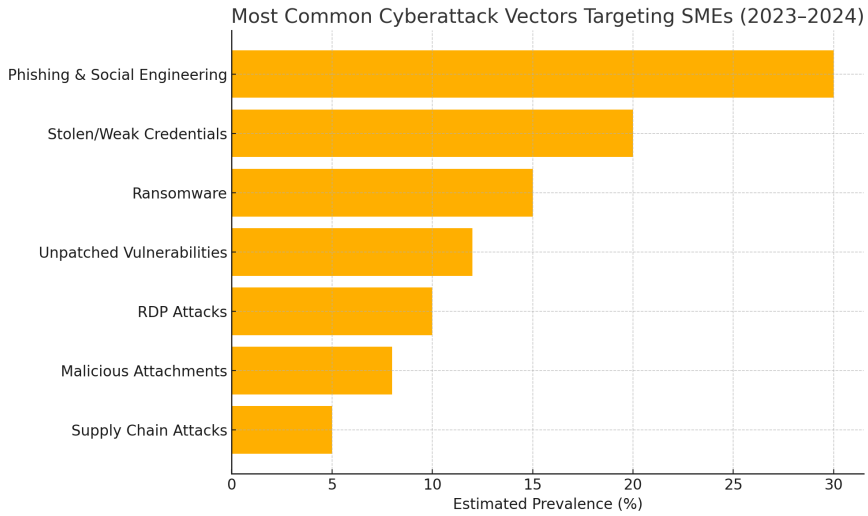


Figure 2. Most common cyberattack vectors targeting SMEs, based on industry reports from 2023–2024. Source: Adapted from Verizon [2–4].

commercial SIEM solutions (such as Splunk, IBM QRadar, or ArcSight) have historically placed them out of reach for SMEs [13]. In many cases, organizations are forced to rely on disparate tools, manual log reviews, or no centralized monitoring at all, resulting in delayed detection and prolonged exposure to threats.

This gap in accessibility shows the importance of evaluating open source SIEM platforms like Wazuh. Wazuh provides a comprehensive set of features including log analysis, intrusion detection, file integrity monitoring, and compliance auditing without the high cost of proprietary tools [14, 15]. For SMEs, Wazuh represents an opportunity to implement enterprise-grade security capabilities on a constrained budget. Additionally, its open source nature enables organizations to customize and extend its functionality, ensuring that the platform can adapt to their specific infrastructure and use cases. By evaluating the performance of Wazuh in real world threat scenarios, this research aims to demonstrate its practicality and effectiveness as a low-cost yet powerful alternative to commercial solutions.

Table 1 highlights the differences between leading commercial SIEM platforms and Wazuh, underscoring its accessibility, flexibility, and suitability for SMEs.

A key consideration in this evaluation is the diversity of endpoint threats that organizations must detect. Attackers continue to exploit vulnerabilities in user devices, operating systems, and

Table 1. Comparison of commercial SIEM platforms and Wazuh. Source: Adapted from Splunk, IBM, ArcSight, and Wazuh product documentation [8–11].

Feature	Splunk	IBM QRadar	ArcSight	Wazuh
Cost (Licensing)	High	High	High	Low / Free
Open Source	No	No	No	Yes
Cloud Compatibility	Yes	Yes	Yes	Yes
Custom Rule Creation	Yes	Yes	Yes	Yes
MITRE ATT&CK Integration	Partial	Yes	Yes	Full
Community Support	Limited	Limited	Limited	Strong
Ease of Deployment for SMEs	Low	Moderate	Low	High

applications, often leveraging fileless malware, scripting engines, and living off the land binaries (LOLBins) to evade traditional defenses. These techniques are particularly difficult to detect, requiring behavioral monitoring, correlation of disparate events, and timely alerting. A modern SIEM must be capable of ingesting and analyzing data from multiple sources Windows event logs, Linux audit logs, Sysmon data, and application logs to identify suspicious patterns across the attack chain. Evaluating Wazuh's ability to detect these diverse endpoint threats is essential to understanding its true value as a defensive tool for SMEs.

Moreover, the appeal of open source tools extends beyond cost savings. Platforms like Wazuh benefit from transparency, community contributions, and integration with threat intelligence frameworks such as MITRE ATT&CK [16]. These attributes promote faster innovation, better documentation, and the ability to audit and modify detection rules to suit an organization's unique risk profile. In contrast to black-box commercial systems, open source platforms empower users to understand and control how alerts are generated, which is particularly valuable for teams looking to build in house security expertise. For SMEs that cannot afford a full security staff, this visibility and control are essential for developing an adaptive and sustainable security posture.

Evaluating Wazuh as an open source SIEM is a critical endeavor especially for organizations that need effective, affordable, and customizable solutions. This research seeks to measure Wazuh's detection capabilities against a range of simulated endpoint threats and assess whether it can deliver the visibility, accuracy, and responsiveness required in today's threat

environment. The findings are intended to help SMEs make informed decisions about adopting open source security technologies and contribute to the broader discussion around scalable, accessible cybersecurity for all.

1.3 Problem Statement

Small to medium sized enterprises (SMEs) are increasingly targeted by sophisticated cyber threats. Despite constituting over 90% of businesses and contributing to 60% of employment worldwide, SMEs often lack the necessary resources and expertise to implement robust cybersecurity measures [18]. This vulnerability is exacerbated by the growing complexity of cyberattacks, which frequently exploit endpoint systems such as workstations, servers, and mobile devices.

Security Information and Event Management (SIEM) systems have emerged as critical tools for monitoring and analyzing security events across organizational networks. Wazuh, an open source SIEM solution, offers features such as log analysis, intrusion detection, and compliance monitoring [19]. Its cost effectiveness and adaptability make it an attractive option for SMEs seeking to enhance their security posture without incurring substantial expenses.

However, there is a notable gap in empirical research evaluating Wazuh's effectiveness in detecting and responding to a diverse array of endpoint based cyberattacks. Existing literature primarily focuses on the architectural components and basic functionalities of open source SIEM solutions, often neglecting comprehensive performance assessments [20]. Consequently, organizations lack concrete data to inform their decision making processes regarding the deployment of Wazuh in real world scenarios. Figure 3 illustrates this research gap, highlighting the lack of empirical studies evaluating Wazuh's detection performance.

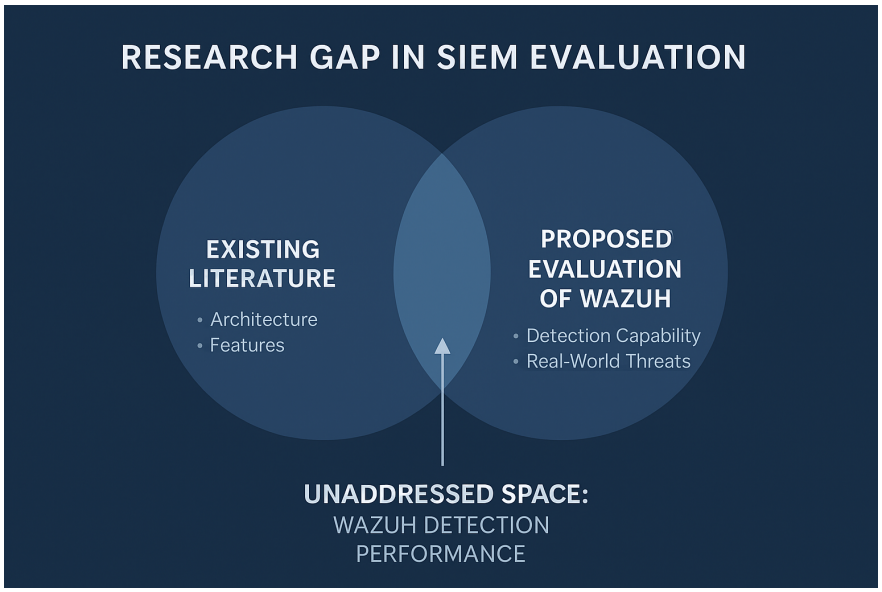


Figure 3. Visual representation of the research gap in Wazuh SIEM evaluation. Existing literature emphasizes architecture and core features, while performance related detection capabilities remain underexplored in real world environments. Source: Adapted from data in IBM [3] and academic reviews [5].

A study published in PLOS ONE highlights this deficiency, emphasizing the need for rigorous evaluations of open source SIEM systems under conditions that simulate actual enterprise environments [21]. The research shows that while open source solutions like Wazuh are gaining popularity, their performance metrics such as detection accuracy, false positive rates, and scalability remain underexplored.

SIEM Research Focus by Area

Area	Existing Literature	Gap Area
Architecture	Well-documented	Underexplored
Functionality	Basic features	Performance evaluation
Detection Accuracy	Underexplored	Underexplored
Scalability	Underexplored	Underexplored

Figure 4. Comparison of current SIEM research focus areas. While architectural aspects are well covered, critical performance metrics such as detection accuracy and scalability remain underexplored in the literature. Source: Adapted from PLOS ONE [5] and IBM [3].

The dynamic nature of cyber threats requires that SIEM systems be capable of detecting a wide spectrum of attack vectors, including but not limited to malware infections, unauthorized access attempts, and data exfiltration activities. The ability to effectively monitor and analyze endpoint activities is paramount, as endpoints often serve as entry points for attackers seeking to compromise organizational networks.

Given these considerations, it is important to conduct a comprehensive evaluation of Wazuh's performance in detecting various endpoint based cyberattacks. This assessment should have diverse operating systems and configurations to reflect the heterogeneous environments typical of SMEs. By addressing this research gap, the study aims to provide valuable insights into Wazuh's capabilities and limitations, helping organizations to make informed decisions about their cybersecurity strategies.

1.4 Objectives

The primary objective of this thesis is to conduct a comprehensive evaluation of Wazuh, an open source Security Information and Event Management (SIEM) platform, in its ability to detect and respond to endpoint based cyberattacks. Using Wazuh generated alert logs collected

from controlled simulations on Windows and Linux endpoints, the research examines Wazuh's effectiveness against common endpoint threat behaviors including brute force login attempts, malware execution behaviors, privilege escalation, and system tampering.

This research examines Wazuh's effectiveness across several MITRE ATTCK tactics relevant to endpoint threats: Initial Access, Credential Access, Lateral Movement, Persistence, Defense Evasion, and Privilege Escalation. In operational terms, these tactics correspond to observable endpoint behaviors such as brute force login attempts (Credential Access / Initial Access), malware execution (Execution, often associated with Initial Access and Defense Evasion), privilege escalation (Privilege Escalation), and system tampering (Defense Evasion and Persistence). The analysis compares Wazuh's detection coverage, timeliness, rule diversity, and severity classifications across these tactics and behaviors on Windows and Linux endpoints.

The study quantifies detection and operational performance using the following indicators: detection coverage (the breadth of MITRE ATTCK techniques represented in alerts), detection timeliness (temporal responsiveness and alert patterns), rule diversity (range and concentration of detection rules triggered), and alert severity classification (consistency and discriminative power of severity assignments). Comparative analysis across Windows and Linux environments will assess cross platform consistency and highlight any OS-specific detection gaps. Findings are intended to inform resource constrained organizations about Wazuh's practical capabilities and to guide configuration or tuning recommendations.

Furthermore, the thesis will compare Wazuh's performance across two operating system environments: Windows and Linux. These platforms represent distinct endpoint architectures and log generation mechanisms, and a comparative analysis will reveal whether Wazuh maintains consistent detection capabilities across them. This objective is particularly significant in hybrid or cross platform environments, where a SIEM's ability to normalize and analyze events across diverse systems is crucial for maintaining unified visibility.

Ultimately, the goal of this research is to generate actionable insights about the strengths and limitations of Wazuh as an open source detection tool. The findings will inform SMEs

and other organizations considering Wazuh about its practical effectiveness and guide future configuration, tuning, or supplemental measures needed to improve endpoint detection and response.

1.5 Research Questions

This study seeks to investigate the capabilities of Wazuh, an open source Security Information and Event Management (SIEM) system, in detecting and responding to simulated endpoint based cyberattacks. Given the increasing reliance on open source security tools by small to medium sized enterprises (SMEs), it is critical to understand how such platforms perform under realistic threat conditions. The research is guided by the following central question:

What is the comparative effectiveness of Wazuh's detection and response capabilities across multiple categories of simulated endpoint attacks?

This primary question is intended to assess not only the breadth of Wazuh's detection rules, but also the timeliness, detection coverage (and proxy measures of accuracy where ground truth is available), and severity of the alerts it produces when faced with different types of adversary behavior. It also invites comparison across distinct endpoint environments to determine whether Wazuh's performance is consistent regardless of operating system or system architecture.

To support the investigation of this core question, the following sub-questions are posed:

How accurately does Wazuh detect various types of endpoint based attacks, such as brute force logins, malware execution, privilege escalation, and system modification? This sub-question addresses the detection fidelity of Wazuh by examining the correlation between simulated attack events and alert generation.

What is the average response time between the initiation of an attack and the generation of a corresponding alert in Wazuh? This question aims to evaluate the efficiency of the system in producing timely alerts that are actionable within a real world security operations context.

How does the diversity of Wazuh detection rules impact the system's ability to recognize distinct techniques and tactics used in endpoint attacks? This considers the range and specificity of detection rules triggered across simulations and their alignment with known adversary techniques

from the MITRE ATT&CK framework.

How do Wazuh's detection and response capabilities vary between Windows and Linux endpoint agents? By comparing detection results across two major operating systems, this sub-question examines whether performance gaps exist based on endpoint type and log source.

What levels of severity does Wazuh assign to different types of attacks, and how well do these severity ratings align with the actual risk posed by each simulated behavior? This explores how Wazuh classifies threats and whether its severity labels support effective prioritization by analysts.

These questions form the investigative framework of the thesis and inform the structure of the analysis. By answering them through systematic testing and evaluation, this study aims to offer meaningful insights into the practical value of Wazuh in modern cybersecurity environments, especially for organizations operating with limited resources.

1.6 Thesis Structure

This thesis is organized into five chapters, each of which contributes to the development, execution, and evaluation of a systematic study on the effectiveness of Wazuh as an open source SIEM tool for detecting and responding to endpoint based cyberattacks. The chapters are structured as follows:

Chapter 1: Introduction: Introduces the research topic and outlines the context, background, and relevance of the study. It discusses the growing complexity of cyber threats, the challenges faced by small to medium sized enterprises (SMEs), and the rise of open source SIEM solutions like Wazuh. The chapter also presents the research motivation, clearly defines the problem, outlines the objectives, and states the central research questions that guide the thesis.

Chapter 2: Literature Review: Reviews existing academic and industry literature related to cybersecurity threats, endpoint attack techniques, SIEM tools, and open source tools. It highlights the evolution of SIEM technologies, compares commercial and open source platforms, and explores frameworks such as MITRE ATT&CK and threat simulation tools like CALDERA. The review also identifies gaps in current research that this thesis seeks to address.

Chapter 3: Methodology: Details the research design and experimental methodology used to evaluate Wazuh. It describes the virtual test environment, including the setup of Windows and Linux endpoints, the configuration of Wazuh agents and server, and the attack simulation process using CALDERA and scripted attacks. This chapter also explains how data was collected, which performance metrics were measured (detection coverage, timeliness, severity), and how the data was analyzed to answer the research questions.

Chapter 4: Results and Discussion: Presents the findings from the controlled endpoint simulations and Wazuh alert logs, and interprets those results in the context of the study's research questions. For each performance indicator — detection coverage, timeliness, severity classification, rule diversity, and cross platform consistency. Presented first is the empirical results (tables and figures) and then discuss of their implications, potential causes, and relevance for small and medium sized enterprises. Methodological details on data preprocessing, MITRE extraction, and severity normalization are summarized here with full code and extended outputs provided in the Appendix.

Chapter 5: Conclusion: The final chapter summarizes the major contributions of the thesis, the importance of evaluating open source SIEM solutions, and reflects on the significance of the findings for cybersecurity decision making. It also restates the research objectives and discusses how they were achieved through the study.

CHAPTER 2

Literature Review

2.1 Overview of SIEM and XDR Platforms

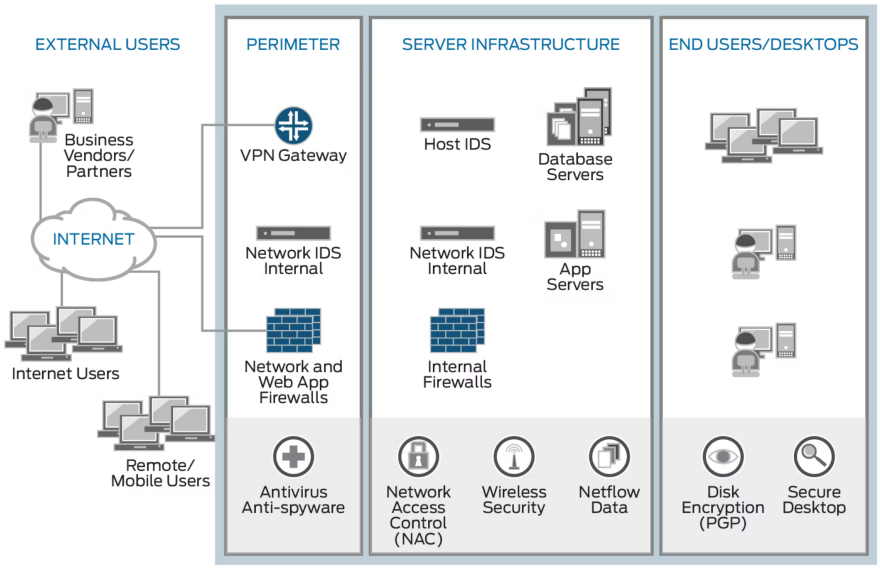


Figure 5. Security architecture showing layered defense across the perimeter, server infrastructure, and endpoints. It highlights data sources, such as IDS, firewalls, and endpoint controls that commonly integrated into SIEM and XDR platforms for centralized threat detection and response [6].

In the evolving landscape of cybersecurity, organizations face increasingly sophisticated threats that require advanced detection and response mechanisms. Two pivotal technologies in this domain are Security Information and Event Management (SIEM) and Extended Detection and Response (XDR) platforms. Both serve as integral components in identifying, analyzing, and mitigating security incidents across complex IT infrastructures.

Security Information and Event Management (SIEM) systems combine Security Information Management (SIM) and Security Event Management (SEM) functionalities to provide a centralized solution for real time security monitoring, alerting, and log management. These

platforms aggregate log data from diverse sources, servers, endpoints, firewalls, intrusion detection systems, and cloud applications and correlate events to uncover malicious activity and compliance violations [22].

Modern SIEM platforms have evolved from basic log collection tools into sophisticated threat detection engines. They now integrate machine learning (ML) and artificial intelligence (AI) to enhance the detection of behavioral anomalies and suspicious patterns [12]. This evolution has enabled SIEM tools to not only detect known threat signatures but also flag deviations from baseline behaviors reducing the mean time to detect (MTTD) and respond to threats. Additionally, SIEMs support forensic analysis and play a central role in incident response workflows, especially in environments subject to regulatory oversight such as HIPAA, PCI-DSS, and GDPR.

Despite these capabilities, traditional SIEMs often present significant barriers to adoption. Their complexity, high licensing fees, and demand for dedicated security teams can be overwhelming for small to medium sized enterprises (SMEs) [20]. Many SMEs lack the technical expertise and financial bandwidth required to deploy, configure, and tune enterprise grade SIEM solutions effectively.

Extended Detection and Response (XDR) represents a newer, more integrated security approach that expands on the visibility and context provided by SIEM. While SIEM systems primarily ingest log data, XDR platforms collect and correlate telemetry across multiple security layers including endpoints, servers, email, identity systems, and network traffic to detect and respond to threats in a unified manner [23].

Unlike traditional tools that require manual correlation across disparate systems, XDR uses automation and advanced analytics to stitch together related security events and prioritize the most severe incidents. It enables faster, broader, and more coordinated detection across the entire attack surface [14]. In doing so, XDR platforms can recognize lateral movement, chained attack techniques, and complex tactics that might otherwise evade detection by siloed security tools.

Another key distinction lies in operational use. While SIEM is often used for compliance

driven monitoring and historical log analysis, XDR is more action oriented focusing on proactive threat hunting, automated response, and real time detection. That said, many modern security architectures integrate both systems to leverage the strengths of each.

SIEM vs. XDR: Complementary roles rather than being mutually exclusive, SIEM and XDR platforms can work in tandem. A well architected security stack may use a SIEM to fulfill logging, auditing, and compliance needs, while XDR handles correlation, enrichment, and automated incident response. Some modern vendors now offer hybrid solutions that embed XDR features into SIEM platforms or vice versa [12, 24].

Understanding the differences and synergies between SIEM and XDR technologies is critical for organizations aiming to strengthen their detection and response capabilities. For SMEs in particular, where budgets are limited and the cost of a breach can be devastating, choosing the right combination of technologies can significantly impact security outcomes.

2.2 Open Source SIEM Tools: Strengths and Limitations

As organizations seek cost effective and adaptable cybersecurity solutions, open source Security Information and Event Management (SIEM) tools have emerged as viable alternatives to commercial platforms. Open source SIEMs offer functionalities similar to those of enterprise grade systems log collection, correlation, alerting, and visualization without the licensing fees associated with commercial solutions. This makes them particularly attractive for small to medium sized enterprises (SMEs).

One of the key advantages of open source SIEMs is their flexibility and extensibility. Solutions such as Wazuh, OSSEC, and Snort allow users to customize rules, integrations, and data processing pipelines to suit their unique environments [15]. This level of configurability supports more granular detection use cases and allows for deeper alignment with frameworks like MITRE ATT&CK [16].

Open source platforms also benefit from active community support. Frequent contributions and transparent development models ensure that these tools are regularly updated with new detection rules and features. For example, Wazuh has evolved from the original OSSEC engine and

has introduced modules for intrusion detection, vulnerability scanning, file integrity monitoring, and compliance auditing [15]. These enhancements make open source SIEMs not only affordable but increasingly robust.

However, these benefits come with tradeoffs. Unlike commercial platforms like Splunk or IBM QRadar, open source SIEMs typically require significant manual configuration, tuning, and ongoing maintenance [20]. Additionally, they often lack native support for advanced analytics or automated incident response unless integrated with third party tools. Many SMEs also struggle with properly deploying and maintaining these solutions due to limited cybersecurity expertise on staff [25].

Security assessments have shown that while open source SIEMs can provide substantial value, their effectiveness depends heavily on the deployment context, rule tuning, and user expertise [20]. As such, organizations must weigh the reduced cost and increased transparency of open source tools against the potential complexity and resource demands of implementation.

Open source SIEM platforms like Wazuh offer a compelling option for organizations seeking customizable and affordable cybersecurity solutions. Their open architectures and community driven development allow for innovation and flexibility, but their successful deployment requires investment in expertise and ongoing configuration.

2.3 Introduction to Wazuh

Wazuh is a prominent open source security monitoring platform that extends the capabilities of its predecessor OSSEC. It functions as both a Security Information and Event Management (SIEM) and an Extended Detection and Response (XDR) tool, offering features such as log data collection, file integrity monitoring, intrusion detection, and compliance auditing [19]. Its architecture and agent based design make it suitable for a wide variety of environments, including on premises, cloud, and hybrid networks.

At its core, Wazuh collects security data from multiple endpoints through lightweight agents installed on various operating systems, including Windows, Linux, and macOS. These agents forward logs and security event data to a centralized Wazuh manager, where the information

is processed and analyzed [20]. The platform can detect a range of threats by applying rules and decoders to the data collected, generating alerts when suspicious activity is identified.

Figure 6 provides an overview of the Wazuh platform's architecture, highlighting the flow of data from agents to the dashboard.

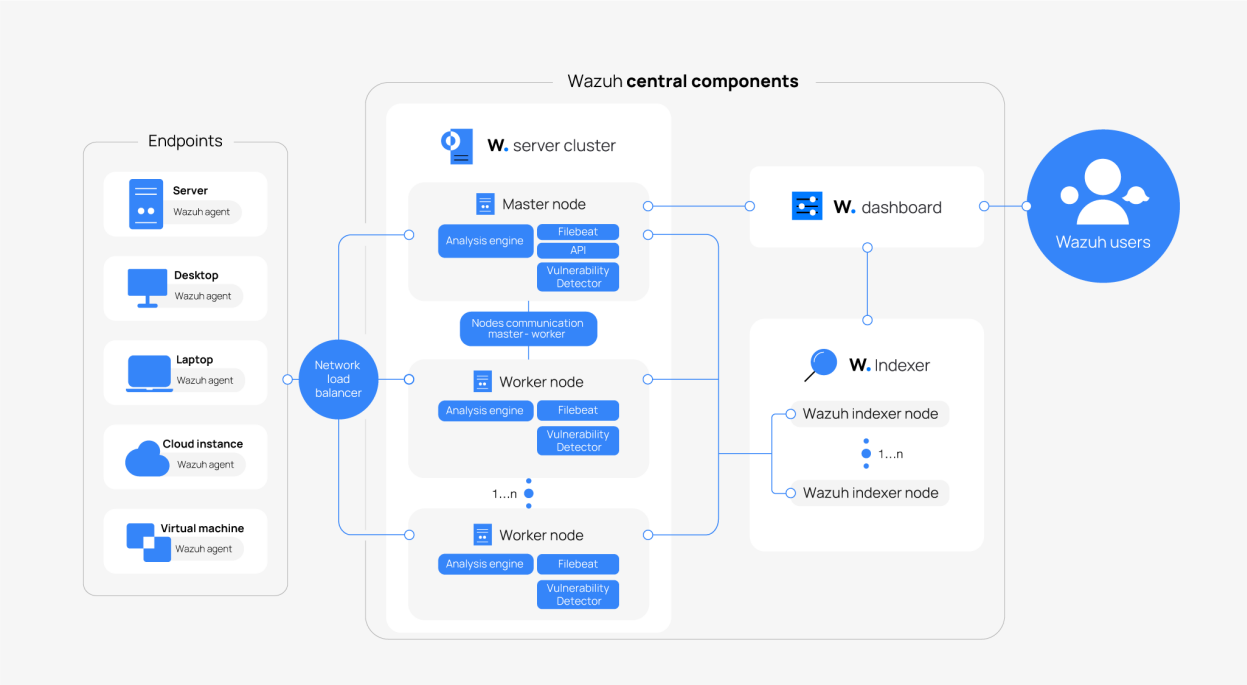


Figure 6. Wazuh platform architecture illustrating the interaction between agents, server, indexer, and dashboard components. Source: Wazuh Documentation [7].

One of Wazuh's key differentiators is its integration with the MITRE ATT&CK framework, which enables mapping of detected behaviors to known adversarial tactics and techniques [21]. This contextualization enhances the quality of threat detection and assists analysts in understanding the nature and impact of potential incidents. Furthermore, Wazuh provides detailed alert metadata including severity levels, rule IDs, and matched signatures, which makes it easier to classify and respond to threats in real time.

Wazuh also supports compliance monitoring for industry standards such as PCI DSS, HIPAA, and GDPR. Its compliance modules automate the auditing of system configurations and log data against specified policies, helping organizations streamline regulatory reporting and maintain security best practices [19].

The platform integrates easily with threat intelligence feeds and third party tools such as VirusTotal, Suricata, and YARA. Wazuh's flexibility enables analysts to enrich detections and customize their monitoring environment to align with specific security goals [16].

Recent evaluations highlight Wazuh's scalability, particularly in environments with diverse endpoint types. However, while the platform offers comprehensive functionality, it requires proper tuning and domain knowledge to optimize detection rules and reduce false positives [17].

Overall, Wazuh's open source nature, robust detection capabilities, and integration with adversarial behavior frameworks position it as a powerful and accessible SIEM/XDR solution for organizations of all sizes.

2.4 Threat Simulation Techniques

Threat simulation is a critical practice in cybersecurity that involves replicating real world adversary tactics, techniques, and procedures (TTPs) in a controlled environment. This process enables organizations to assess the effectiveness of their detection and response mechanisms, identify coverage gaps, and improve their incident response strategies. Unlike traditional penetration testing, which focuses on exploiting vulnerabilities, threat simulation emphasizes visibility and detection, often aligned with threat frameworks such as MITRE ATT&CK [26].

One of the most widely adopted threat simulation frameworks is the MITRE ATT&CK framework, which provides a comprehensive matrix of adversary behaviors based on real world observations. Tools that integrate ATT&CK enable organizations to simulate specific techniques across various stages of the cyber kill chain, including initial access, execution, persistence, and exfiltration. These simulations help security teams map out their detection capabilities and tailor defenses to address the most relevant threats [26].

Threat simulation frameworks are essential for evaluating the effectiveness of security monitoring and detection capabilities in realistic, controlled environments. Atomic Red Team, developed by Red Canary, is one such open source framework that has gained popularity for its accessibility and modular structure. It provides a library of YAML based atomic tests that align directly with the MITRE ATT&CK framework, enabling security teams to execute individual

atomic tests representing discrete TTPs [27]. These tests can be safely executed in production like environments and are particularly valuable for validating endpoint detection and response systems such as Wazuh. For example, a security team may simulate a credential dumping technique on a Windows endpoint to verify whether the SIEM correctly detects the behavior, generates an appropriate alert, and assigns a meaningful severity score [28].

While ART focuses on individual techniques, CALDERA, developed by MITRE, offers a more comprehensive adversary emulation approach. CALDERA provides a plugin based architecture capable of chaining multiple ATT&CK mapped abilities into multistage campaigns, closely mirroring realistic attacker behavior [29,30]. By leveraging its builtin Sandcat implant and automated campaign orchestration, security teams can test SIEM rule coverage and detection logic at scale, across different endpoint types and operating systems. This enables researchers to identify platform specific detection gaps, tune alerting thresholds, and improve overall detection fidelity. In this study, CALDERA was chosen as the primary threat simulation platform to support cross platform adversary emulation, while ART serves as an example of more targeted technique level testing.

The adoption of adversary emulation in cybersecurity strategy is also growing in line with the demand for threat informed defense. Red and blue teams now collaborate more closely through purple teaming exercises, in which threat simulation is paired with realtime detection validation. This approach promotes iterative tuning of detection logic and encourages security maturity across the enterprise [31].

However, effective use of threat simulation tools requires both technical skill and environmental readiness. Improper execution of simulations or a lack of visibility into endpoints may lead to inaccurate results or missed detections. It is therefore critical to prepare and monitor the environment closely during simulated attack campaigns to ensure meaningful analysis [26].

In the context of open source SIEM evaluation, threat simulation offers a cost effective, repeatable, and scalable method for stress testing detection capabilities. This thesis leverages CALDERA and attack scripts to simulate various endpoint attacks and analyze how well Wazuh

can detect and respond across different operating systems and agent configurations.

2.5 Previous Research and Gaps

The evaluation of SIEM tools, especially open source platforms like Wazuh, is an area of growing academic and operational interest. Numerous studies have explored the functionality and architecture of SIEM systems, yet there remains a lack of systematic research that examines how well these systems detect diverse endpoint threats in realistic conditions. While commercial tools such as Splunk and IBM QRadar are widely covered in literature, open source solutions often receive less rigorous analysis [20].

Several studies have investigated the theoretical and technical capabilities of Wazuh. Manzoor et al (2024) evaluated open source SIEM platforms in the context of budget constrained small to medium sized enterprises (SMEs). Their findings highlighted Wazuh's competitive advantage in flexibility, rule customization, and cost efficiency compared to commercial tools. However, the study acknowledged that Wazuh's practical detection capabilities across multiple operating systems and attack categories still need further examination [20].

Similarly, Rombaldo Junior et al. (2023) conducted a comprehensive review of cybersecurity readiness in SMEs, emphasizing that most lack the financial resources and technical expertise to deploy enterprise grade security systems [25]. Their review called for increased development and evaluation of affordable, scalable SIEM options such as Wazuh. However, their work remained largely conceptual, with limited experimentation involving real attack simulations or endpoint analysis.

Additionally, although Wazuh documentation and white papers provide architectural details and configuration guidelines, there is little peer reviewed analysis on how well Wazuh performs under active attack scenarios such as malware execution, brute force logins, or lateral movement [32]. This gap presents a major opportunity for applied research.

The role of threat simulation in SIEM testing has been addressed in various purple teaming and adversary emulation studies. However, these studies predominantly focus on commercial EDR/XDR platforms and neglect to systematically assess open source SIEM tools in similar

conditions. Moreover, most comparative evaluations stop at performance benchmarks (CPU or memory usage) rather than detection accuracy, rule effectiveness, or cross platform consistency [33].

This thesis aims to fill these gaps by simulating real world endpoint attacks using CALDERA and scripted attacks and measuring Wazuh's response across Windows and Linux endpoints. The study uniquely evaluates detection coverage, rule diversity, and timeliness factors that are often overlooked in current academic literature. In doing so, it provides empirical evidence on the practical detection performance of Wazuh, especially within environments constrained by budget and resources.

CHAPTER 3

Methodology

3.1 Research Design

This study employs a simulation based experimental design to evaluate the detection and response capabilities of Wazuh, an open source Security Information and Event Management (SIEM) and Extended Detection and Response (XDR) platform. The primary objective is to assess Wazuh's effectiveness in identifying and responding to various endpoint attack techniques across different operating systems.

The experimental design is structured around an Ubuntu 22.04 LTS host machine, provisioned with an Intel Core i7-9700 CPU, 16 GB RAM, and a 512 GB SSD. On this host, CALDERA (v4.2.0) is installed directly to orchestrate adversarial simulations and manage agent deployment. The host also runs the KVM/QEMU hypervisor, which virtualizes the rest of the environment. Two endpoint VMs are provisioned, one running Windows 10 Pro and the other Debian 12, each with the Wazuh agent installed, configured, and registered to the Wazuh manager for continuous log forwarding and response. A third control VM mirrors the Debian endpoint but operates without Wazuh to provide a baseline of unmonitored attack activity.

The Wazuh manager (v4.7.1), with Elasticsearch and Kibana integrations, is deployed in a dedicated Ubuntu VM and serves as the central monitoring and response engine. All components are networked within a private 192.168.1.0/24 subnet, isolated through iptables rules, with CALDERA executing attack profiles against the endpoints and Wazuh collecting, analyzing, and responding to the resulting events. The design follows a pretest intervention, post test structure: endpoints and manager are prepared and baseline connectivity validated, CALDERA executes scripted attacks, and Wazuh detection/response outputs are then collected and analyzed. Randomness is controlled with fixed seeds, and three independent simulation runs are performed for variability, while ethical safeguards ensure that all simulations are restricted to nondestructive

activities within the controlled environment.

Although this study did not utilize the assessment architecture to evaluate Internet of Things (IoT) devices, the same framework can be extended for that purpose. By incorporating IoT devices as additional endpoints within the system, researchers can apply the same detection and evaluation metrics to assess SIEM effectiveness in IoT environments, thereby broadening the applicability of the architecture.

3.2 Environment Configuration

The environment begins with the Ubuntu 22.04 LTS host, which functions both as the physical foundation for virtualization and as the platform where CALDERA is installed and operated. CALDERA is configured with its HTTP C2 server bound to port 8888 and manages Sandcat agents installed on the endpoint VMs, orchestrating adversarial techniques through its Atomic planner. Using KVM/QEMU, a dedicated VM is provisioned for the Wazuh manager, deployed with version 4.7.1, fully integrated with Elasticsearch 8.2 and Kibana 8.2 for log storage and visualization.

Wazuh's MITRE ATT&CK detection rules are configured in var/ossec/etc/rules/local_rules.xml, and active response is enabled in /var/ossec/etc/ossec.conf, with manager agent communications established over ports 1514 (UDP), 1515 (TCP), and 55000 (API). Two additional VMs are provisioned as endpoints: a Windows 10 Pro VM and a Debian 12 VM, each with 2 vCPUs, 2 GB RAM, and 50 GB storage. On Windows, Windows Defender is disabled to avoid interference, while Debian uses a ufw firewall limited to simulation traffic. A control Ubuntu VM mirrors the Debian configuration but runs without the Wazuh agent, serving as a baseline.

Networking is isolated within a 192.168.1.0/24 subnet, with iptables restricting access to Wazuh and CALDERA ports. SELinux/AppArmor are placed in permissive mode to avoid disrupting controlled simulations. Baseline checks include ping and nc connectivity tests, as well as vulnerability scans, ensuring that the environment is stable and reproducible prior to executing attacks.

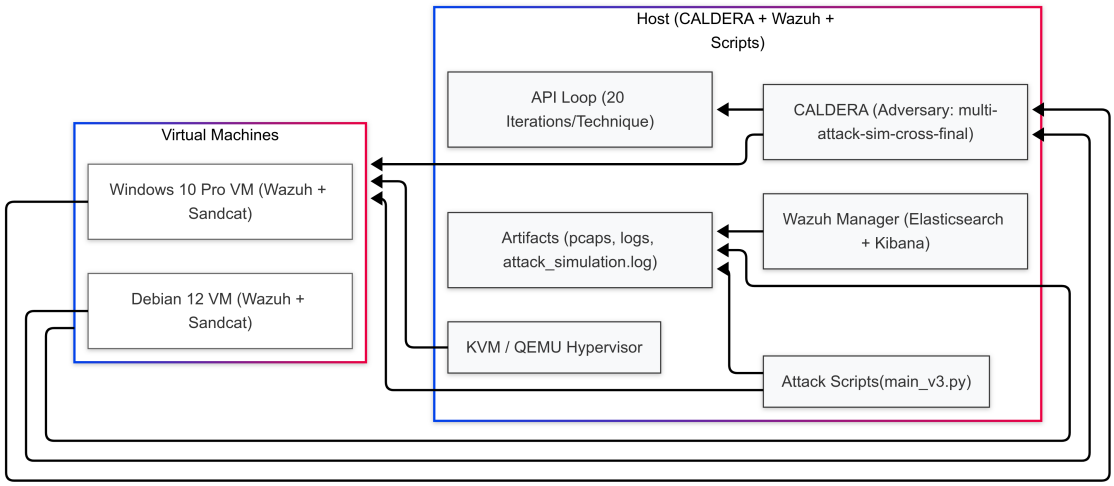


Figure 7. Experimental environment architecture. The setup integrates CALDERA for adversary emulation, the Wazuh manager (Elasticsearch + Kibana), and custom attack scripts (main_v3.py) on the host system, coordinated through an API loop executing 20 iterations per MITRE ATT&CK technique. Two monitored virtual machines (Windows 10 Pro and Debian 12) run both Wazuh agents and the Sandcat implant for adversary emulation. The KVM/QEMU hypervisor provisions the VMs, while artifacts such as pcaps, logs, and attack_simulation.log are collected for evaluation. This architecture enables controlled, repeatable, and cross-platform endpoint attack simulations to assess Wazuh's detection and response effectiveness.

3.3 Attack Simulation

The attack simulation is designed as a controlled, repeatable, and fully instrumented campaign that exercises a broad set of endpoint threat categories while preserving nondestructive behavior and strict lab containment. CALDERA, installed on the host, acts as the orchestration engine, the campaign is driven by a coordinating controller script, (main_v3.py) that interfaces with CALDERA's APIs and local scheduling, and also logs every planned and executed action to a ground-truth file (attack_simulation.log). Each simulation run is composed of timed phases and subphases so that adversary activity is predictable, measurable, and safely rate capped: a warm up period of 30 to 60s to validate connectivity and agent heartbeats, an execution window of 5 to 10 minutes when attack techniques are executed, and a cool down and observation window (greater than 5 minutes) to capture delayed alerts and active responses.

Orchestration and sequencing: the campaign uses an execution plan that groups techniques into logical waves to avoid resource starvation on endpoints and to make signal attribution clear. Each wave targets a subset of techniques (authentication focused wave, reconnaissance wave,

exploitation simulation wave, DoS stress wave). Waves are scheduled with configurable interwave delays and intratechnique jitter to emulate realistic attacker timing: the controller enforces a minimum interaction delay (configurable, 100–500 ms) and a jitter percentage (±30%) so that request timing is not perfectly periodic. All randomization is needed for reproducibility across runs.

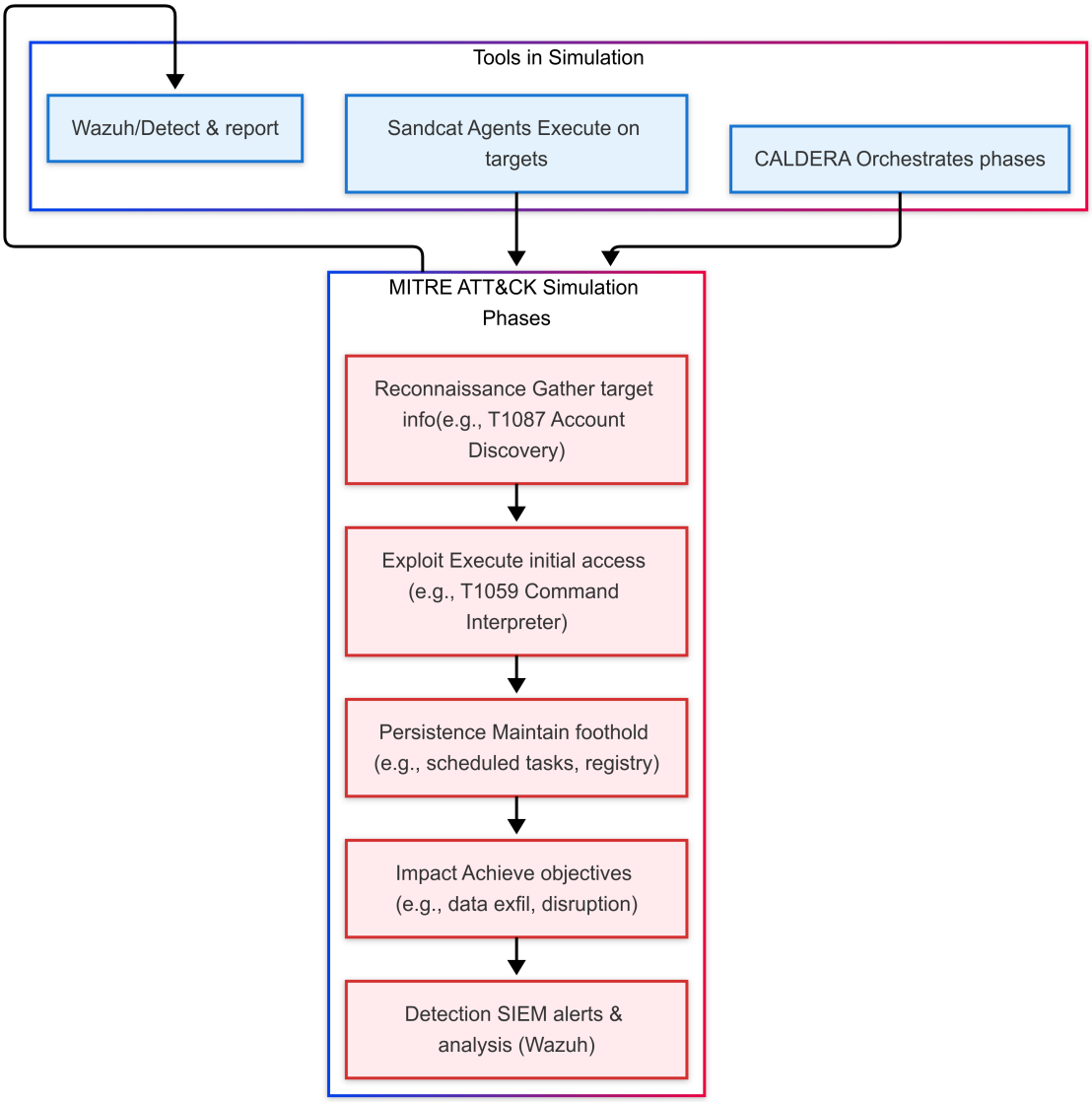


Figure 8. MITRE ATTCK simulation phases with tool mapping (CALDERA orchestration, Sandcat execution, Wazuh detection). The diagram illustrates a single simulated campaign flow from reconnaissance through impact and into SIEM detection and logging

Technique selection and safe emulation: techniques are mapped to MITRE ATT&CK

identifiers and implemented as behavioral simulations rather than exploitative attacks. For each technique the campaign defines (1) intent (what attacker behavior is being emulated), (2) observable telemetry (the log lines, network patterns, or file system changes expected), (3) safe action (the benign operation used to create the telemetry), and (4) rate/resource caps to avoid destabilizing VMs.

Examples of technique classes and their safe emulation approach:

- **Credential access (brute force style simulation):** emulate repeated authentication attempts using the system's authentication interface but with limited attempt rates and invalid credentials. Observable telemetry includes repeated authentication failure entries in system and authorization logs and repeated failed session establishment attempts at the network layer. The experiment configures maximum attempts per minute and enforces exponential backoff on repeated failures so the endpoint is not overwhelmed.

- **Discovery (account & network enumeration):** emulate benign discovery commands that surface the same artifacts attackers seek (user and network configuration queries) but run with minimal privilege and read only operations. Expected telemetry includes presence of user enumeration commands in command history logs and network configuration outputs in system logs. Timing and frequency are limited to reduce noise.

- **Execution (scripting/command activity):** emulate command interpreter usage by executing timestamped, benign echo or no-op commands that create clear execution events in process creation logs and shell history without changing system state. Each execution is accompanied by a small marker string in stdout/stderr that the controller logs to attack_simulation.log to enable high confidence correlation to capture delayed alerts and active responses.

- **Reconnaissance (port scanning behavior):** emulate scanning by issuing sparse probes across a controlled port list with explicit per port inter probe delays and upper limits on concurrent sockets to avoid service disruption. The probes are designed to generate

characteristic TCP/UDP connection attempts without establishing stateful sessions. To be expected: repeated connection attempts, RSTs, or ICMP unreachable messages logged by the kernel or firewall.

- **Denial of Service (non destructive stress patterns):** emulate load patterns by sending low volume, patterned traffic that exercises detection rules for volumetric or resource exhaustion behavior but is constrained by fixed socket caps, per second packet limits, and payload size caps. Examples include short bursts of SYN style connection attempts and HTTP request bursts with enforced cool downs between bursts. Packet rates are set well below levels that would crash services and an automated cutoff cancels a technique if CPU, memory, or I/O usage on any endpoint exceeds a configured threshold (for example 60% CPU, 70% RAM).

- **Application layer injections (simulated SQLi/XSS indicators):** rather than deploying active exploits, the simulation submits non malicious indicator strings to application endpoints or test pages that will generate the same logging artifacts (for example, special tokenized strings placed in query parameters or headers). These tokens are selected so that they are logged verbatim by the application or web server and recorded in the attack simulation log, enabling correlation without executing dangerous payloads.

Technique parameters and concurrency: the controller uses a thread pool max workers=3 to run parallel tasks while limiting per technique concurrency. Each technique run uses a per task configuration block that includes iteration count (20), per iteration delay range, timeout policy, and failure handling. If a technique fails repeatedly (connection refused or unexpected exception), the controller records the failure, applies a backoff, and continues to the next item to avoid cascading failures. All attempts are instrumented with unique correlation IDs and recorded in JSON lines format in attack simulation log with fields.

Telemetry expectations and mapping to Wazuh artifacts: for each technique the simulation defines the expected signal set and where it should appear in Wazuh's pipeline. Examples of mapped observables (described at a detection signal level, not as exploit code) include:

repeated “authentication failure” entries in syslog or Windows Security event logs for brute force simulations, multiple shortlived TCP SYNs with no complete handshake for scanning waves, sudden increases in process creation events or command interpreter entries for execution simulations, application access logs containing the test tokens for injection simulations, and elevated connection/packet rate logs or kernel level resource warnings for DoS waves. These mappings are also documented in an internal “expectation matrix” that the post test correlation uses to link Wazuh alerts back to ground truth.

Logging, ground truth, and validation: robust ground truth capture is critical. The controller writes every planned action and runtime outcome to attack simulation log as JSON lines, and the host simultaneously captures a short pcap (network capture) and endpoint process snapshots for each wave. Each filesystem or command output artifact that is intended to signal an attack includes a unique marker string so the post test correlation is deterministic. All systems are synchronized to NTP before experiments to ensure timestamp alignment; the post test pipeline performs temporal joins between attack simulation log, Wazuh event exports, and pcap or endpoint logs within a configurable correlation window (±10 s by default) to establish linkage.

Monitoring, safety cutoffs, and rollback: the controller continuously monitors endpoint health (CPU %, memory %, disk I/O, network throughput) and listens for any kernel or hypervisor-level alerts. If any safety threshold is crossed, the controller aborts the current technique, escalates to an administrative halt for the run, and triggers VM snapshot rollback procedures if required. Pre run snapshots are taken for all VMs so any unintended changes can be reverted instantly. In addition, the experiment enforces a conservative resource policy: per technique maximum concurrent sockets, minimum inter request delay, maximum runtime per technique, and overall runtime budget per wave. The test plan includes explicit “kill switches” and requires operator confirmation to restart aborted runs.

Auditability and repeatability: to ensure reproducibility, the full simulation configuration (technique parameters, wave schedules, jitter settings, and resource caps) is stored alongside attack simulation log and a human readable run manifest. Each run is given a deterministic identifier

derived from the seed, timestamp, and manifest hash. Post run, the correlation pipeline produces a machine readable mapping file that pairs each ground truth action with the matching Wazuh event IDs and log snippets (where matches exist), plus a "no match" list for actions that produced no observable Wazuh alerts.

Post test artifacts and analyses collected: alongside Wazuh event exports and attack simulation.log, the campaign captures: endpoint process accounting summaries, selective log extracts (auth logs, application access logs, syslog/Windows event logs), short pcaps for each wave, and the Wazuh active responses log entries. These artifacts enable multiaxis analysis, timeline reconstruction, per technique signal analysis, and deterministic rule tuning, without needing exploit code. All artifacts are stored in a locked project workspace with access controls and documented retention/secure deletion policies.

Ethics, containment, and documentation: every simulation run is executed under an approved IRB or lab protocol, runs only within the isolated 192.168.1.0/24 subnet, and uses nondestructive, benign commands or marker tokens. The plan documents operator roles, escalation paths, and emergency rollback procedures. Before first execution, all target VMs are snapshotted and baseline health and service availability are verified.

3.3.1 Campaign Manifest Structure To ensure experimental repeatability and traceability of simulated attack activity, a structured **campaign manifest** was generated for each adversary emulation run. Each manifest entry corresponds to a single executed attack stage and contains:

- **Campaign and Iteration IDs** — identifying the simulation run and stage order.
- **MITRE ATT&CK Technique** — mapped or inferred from the executed ability.
- **Stage Description** — textual description of the technique or command.
- **Timestamp (ISO 8601)** — precise execution time of the stage.
- **Target OS and Agent** — the platform and system where the action occurred.
- **Expected Artifact Fields** — Wazuh log attributes expected for detection.

This manifest acts as the **ground truth timeline** of adversarial actions, against which Wazuh detections are compared. It supports both temporal correlation (alert time vs. execution

time) and technique coverage analysis.

Table 2 summarizes the distribution of attack stages across Windows and Linux endpoints. The full manifest, including stage identifiers, technique mappings, and command details, is provided in **Appendix A.2.1**.

Each manifest entry corresponds to a single executed *link* (an atomic execution step) and includes execution metadata. For transparency and reproducibility, the full manifest CSV files (archived as supplementary materials) include the raw `Link Command` and `Plaintext Command` fields for every link. These command fields record the exact payload executed (for example encoded PowerShell or shell pipelines) and were used, within an isolated laboratory environment, as ground truth for correlating execution events with Wazuh detections. To preserve safety and readability, a summarized excerpt of the manifest (omitting full command payloads) is presented in Appendix A.2.1, while the redacted full CSVs are provided in the supplementary data folder.

Table 2. Summary of Campaign Manifest by Operating System

Target OS	Iterations	Total Stages	Distinct Techniques
Windows	3	80	12
Linux	3	95	11

Each iteration represents a discrete execution burst separated by defined time intervals, enabling temporal segmentation of Wazuh alerts during analysis. This structure also allows for platform-specific evaluation of detection latency, rule coverage, and severity classification.

3.4 Data Collection

Data collection centers on Wazuh's centralized logging, with endpoint agents continuously forwarding logs to the Wazuh manager VM at five second intervals. The Wazuh REST API (/api/3.0/events) is used during and after simulations to pull structured event data, while Elasticsearch provides indexed storage and Kibana supports visualization. To mitigate indexing delays, a five minute buffer period follows each simulation before data extraction. The dataset is structured in JSON with fields such as event_id, timestamp, rule_id, severity, mitre_tactic,

description, allowing fine grained analysis.

Synchronization is enforced with NTP aligned timestamps across host and VMs, and data integrity is preserved through duplicate elimination and manual cross verification with attack_simulation.log. Events are mapped to MITRE ATT&CK categories through Wazuh's rule IDs (brute force detections tagged with Rule ID 100600), while active response actions are correlated using /var/ossec/logs/active-responses.log, which records the precise time of mitigations. Reviewer validation of detections against attack ground truth ensures classification accuracy, and interrater reliability is maintained above Cohen's kappa 0.8. Across three runs, the environment produces approximately 300–500 events, generating a comprehensive record of Wazuh's behavior under CALDERA driven adversarial activity.

3.5 Metrics for Evaluation

3.5.1 Metrics for Evaluation The evaluation framework is grounded in quantitative performance metrics derived from repeated adversarial simulations and benchmarked against a ground truth event log (attack_simulation.log). Metrics capture detection effectiveness and operational efficiency and are reported per MITRE ATT&CK technique and per endpoint (Windows, Linux), with aggregated means and 95% confidence intervals over 20 iterations. A compact summary appears in Table 3.

3.5.1.1 Precision, Recall, F1 Alerts are labeled relative to ground truth as true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN). We report Precision, Recall, and F1:

$$\text{Precision} = \frac{TP}{TP+FP}, \qquad \text{Recall} = \frac{TP}{TP+FN}, \tag{3.1}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3.2}$$

3.5.1.2 Detection Timeliness Let t_{event} denote the timestamp of an executed action from attack_simulation.log and t_{alert} the first corresponding Wazuh alert. Latency is

$$\Delta t = t_{\text{alert}} - t_{\text{event}}. \tag{3.3}$$

We analyze the empirical distribution of Δt (mean, median, variance) and its CDF to capture tail behavior. For operational interpretation, we stratify:

- **Immediate:** $\Delta t < 5$
- **Near real-time:** $5 \leq \Delta t \leq 60$
- **Delayed:** $\Delta t > 60$

However, in this study, the lack of precise alert timestamps in the exported Wazuh event dataset prevented the direct calculation of detection latency values. Although defined as part of the intended evaluation framework, the metric was not computed for the presented results. Instead, the ground-truth timeline was used to establish the temporal structure of the attack sequences, leaving latency measurement as an area for future work.

3.5.1.3 Detection Coverage Coverage quantifies the breadth of adversarial behavior observable by Wazuh:

$$\text{Coverage} = \frac{N_{\text{techniques\_detected}}}{N_{\text{techniques\_executed}}}. \tag{3.4}$$

We further decompose coverage per technique into (i) *rule activation diversity D_r* (count of distinct rules triggered), (ii) *technique mapping fidelity M_t* (proportion of alerts correctly mapped to the intended MITRE identifier), and (iii) *cross platform breadth B_p*:

$$B_p = \frac{|C_{\text{win}} \cap C_{\text{lin}}|}{|C_{\text{win}} \cup C_{\text{lin}}|}, \tag{3.5}$$

where C_{win} and C_{lin} are the sets of covered behaviors on Windows and Linux, respectively.

3.5.1.4 Cross Platform Consistency For metric $m \in \{\text{Accuracy}, \text{Precision}, \text{Recall}, \text{Severity}\}$, platform consistency is

$$\text{Consistency}(m) = 1 - \frac{|m_{\text{win}} - m_{\text{lin}}|}{\max\{m_{\text{win}}, m_{\text{lin}}\}}. \tag{3.6}$$

Values approaching 1 indicate high alignment across platforms.

3.5.1.5 Severity Classification Let S_{obs} be the observed Wazuh severity and S_{exp} the expected label derived from technique criticality and scenario context. The severity error rate is

$$E_s = \frac{1}{N}\sum_{i=1}^{N} 1\left(S_{\text{obs},i} \neq S_{\text{exp},i}\right), \tag{3.7}$$

with stratification into *underestimation* ($S_{\text{obs}} < S_{\text{exp}}$) and *overestimation* ($S_{\text{obs}} > S_{\text{exp}}$).

3.5.1.6 Alert Volume and Noise Ratio Let A_t denote the alert count for technique t. Operational signal quality is summarized via

$$\text{SNR} = \frac{TP}{TP+FP}, \tag{3.8}$$

where values near 1 indicate high signal fidelity with minimal noise. We also inspect per-technique alert distributions to identify redundancy and analyst load implications.

Table 3. Evaluation Metrics Definitions and Equations

Metric	**Definition**	**Equation / Expression**
Detection Accuracy	Proportion of correctly identified events relative to all outcomes.	$Acc = \frac{TP+TN}{TP+TN+FP+FN}$
Precision	Proportion of alerts that are true positives.	$Precision = \frac{TP}{TP+FP}$
Recall	Proportion of malicious events that are correctly detected.	$Recall = \frac{TP}{TP+FN}$
F1-score	Harmonic mean of precision and recall.	$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$
Detection Timeliness Categorized: Immediate $<$ 5s; Near Real-Time 5–60s; Delayed $>$ 60s.	Latency between event occurrence and alert generation.	$\Delta t = t_{alert} - t_{event}$
Detection Coverage	Extent of adversarial techniques detected by Wazuh.	$Coverage = \frac{N_{techniques\_detected}}{N_{techniques\_executed}}$
Rule Activation Diversity	Number of distinct rules triggered per technique.	$D_r = \lvert R_{technique} \rvert$
Cross-Platform Breadth	Overlap of techniques detected across Windows and Linux.	$B_p = \frac{\lvert C_{win} \cap C_{lin} \rvert}{\lvert C_{win} \cup C_{lin} \rvert}$
Cross-Platform Consistency	Alignment of metric values across platforms.	$Consistency(m) = 1 - \frac{\lvert m_{win} - m_{lin} \rvert}{\max(m_{win}, m_{lin})}$
Severity Error Rate	Misclassification of severity relative to ground truth.	$E_s = \frac{1}{N} \sum_{i=1}^{N} 1(S_{obs,i} \neq S_{exp,i})$
Signal-to-Noise Ratio (SNR)	Ratio of meaningful detections to total alerts.	$SNR = \frac{TP}{TP+FP}$

3.6 Tools for Analysis

Data analysis and visualization were performed using Python 3.11 on macOS and Linux environments. The primary libraries utilized included pandas for data cleaning and transformation, matplotlib and eaborn for statistical plotting, and numpy for numerical operations. Data were exported to Excel format for tabular review and cross verification. Log parsing and correlation between Wazuh alert data and ground truth logs were automated through custom Python scripts developed in Visual Studio Code.

All statistical calculations, including detection rate, latency, and severity frequency distributions, were executed using Jupyter Notebook environments. Visualization outputs, such as bar plots, violin plots, and heatmaps, were generated to represent trends across operating systems and attack techniques. These visualizations were later embedded in Chapter 4 to illustrate key findings.

The combination of open source Python packages and structured log data ensured transparent, reproducible, and cross platform analysis of Wazuh's detection and response performance.

CHAPTER 4

Results and Discussion

4.1 Alert Volume and Frequency by Attack Type

Ability Name	attempts	detected
Custom - Encoded PowerShell Test (110000)	12	12
SUDO Brute Force - Debian	7	7
Access /etc/passwd (Local)	6	6
Account Discovery (targeted)	6	6
Cached Credential Dump via Cmdkey	6	6
Delete Windows Defender Scheduled Tasks	6	6
Dump history	6	6
Prevent Powershell History Logging	6	6
Remote System Discovery - arp nix	6	6
emacs spawning an interactive system shell	6	6
iwr or Invoke Web-Request download	6	6
WMI Reconnaissance List Remote Services	5	5
Clear Bash history (cat dev/null)	3	3
Packet Capture Linux using tshark or tcpdump	3	3

Table 4. Summary of simulated adversary abilities executed during the evaluation, including attempt and detection count.

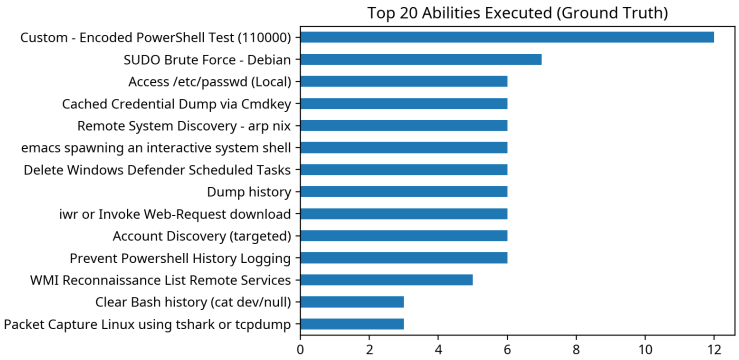


Figure 9. Top 20 techniques executed during the adversary emulation campaign based on ground truth logs. The encoded PowerShell test and SUDO brute force attacks show the highest frequency.

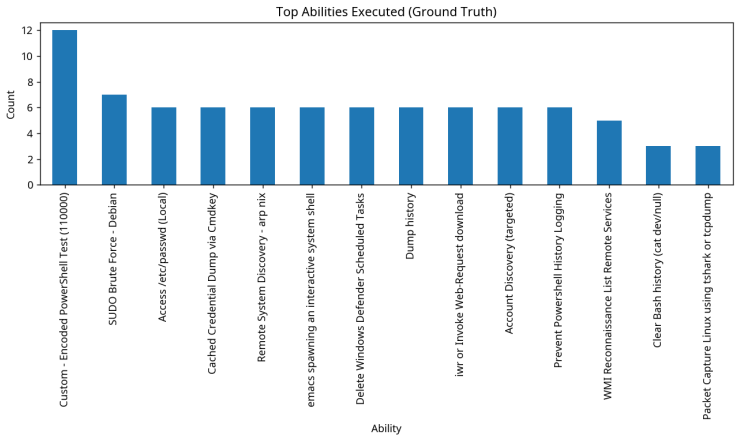


Figure 10. Frequency of top executed adversary abilities during the experiment. The encoded PowerShell test and SUDO brute force attempts dominated execution events, forming the core dataset for evaluating Wazuh's detection and correlation accuracy.

4.2 Detection Performance by Agent

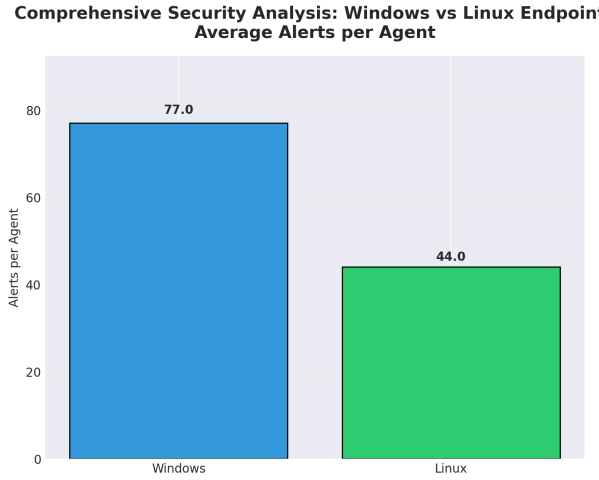


Figure 11. Comparison of average alerts generated per agent across Windows and Linux endpoints. Windows hosts produced a higher alert volume, suggesting more extensive rule coverage and event correlation.

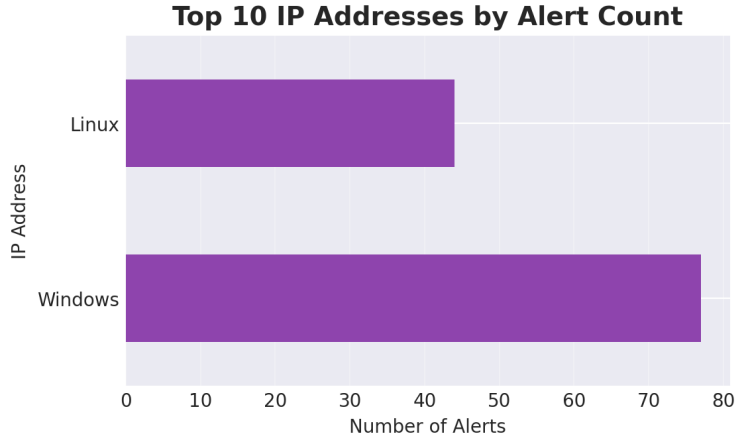


Figure 12. Total number of alerts generated per operating system. Windows produced 77 alerts while Linux generated 44, reflecting higher event correlation and rule triggering on Windows endpoints.

4.3 Rule Diversity and Severity Analysis

```
--- Alert Severity Distribution ---
```

	Windows	Linux
3	3	0
5	10	0
7	10	12
8	6	0
10	16	32
11	3	0
12	8	0
13	4	0
15	17	0

Table 5. Distribution of alert severities generated by Wazuh during adversarial simulations. Severity values correspond to Wazuh's internal classification scale. Linux alerts were concentrated at severity 10 (n = 32) and severity 7 (n = 12), reflecting high impact detections clustered at a limited set of severity levels. Windows alerts were distributed more broadly across severity levels 3–15, with notable peaks at severity 15 (n = 17) and severity 10 (n = 16). This pattern suggests platform specific differences in rule activations and classification intensity.

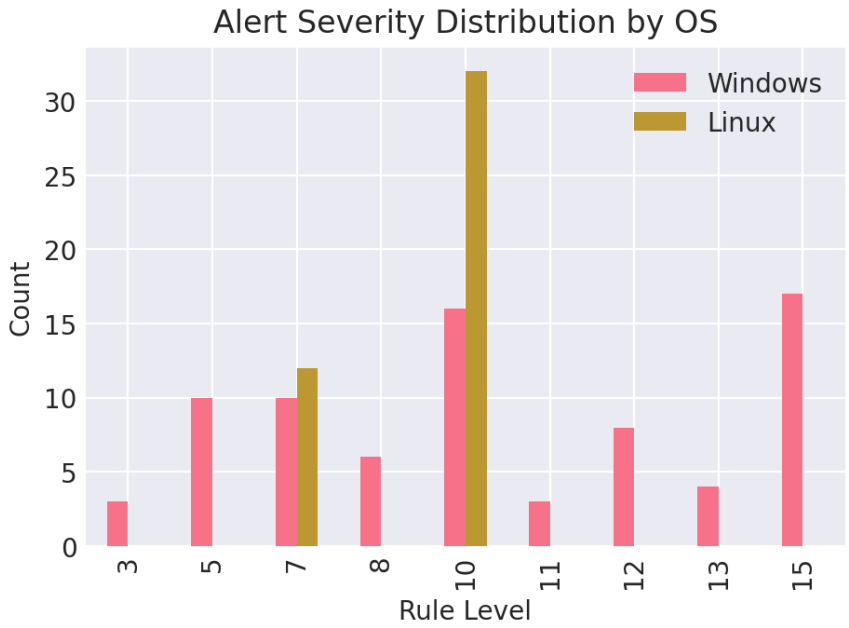


Figure 13. Alert severities were concentrated at specific levels, with Severity 10 representing the largest share of Linux alerts (32) and Severity 15 representing the largest share of Windows alerts (17). The distribution suggests a heavier concentration of high severity alerts for Linux, reflecting rule matches tied to more critical detections. Windows exhibited a broader spread of severities, indicating more diverse rule activations. This descriptive severity analysis provides operational context for interpreting detection patterns in the absence of expected severity baselines.

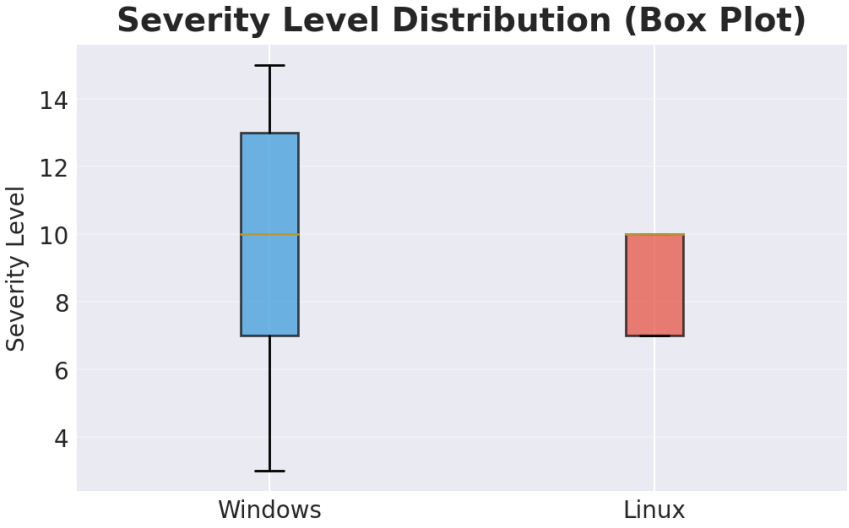


Figure 14. Box plot illustrating the dispersion of severity levels. Windows exhibits greater variability, while Linux displays a narrower interquartile range centered around high severity values.

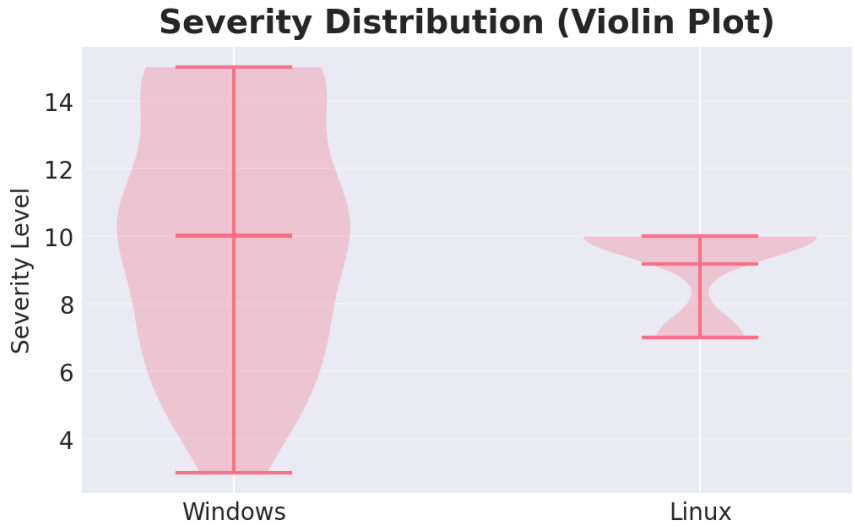


Figure 15. Violin plot showing the density of alert severities. The distribution for Windows is multi modal with several peaks, while Linux maintains a more stable and concentrated profile.

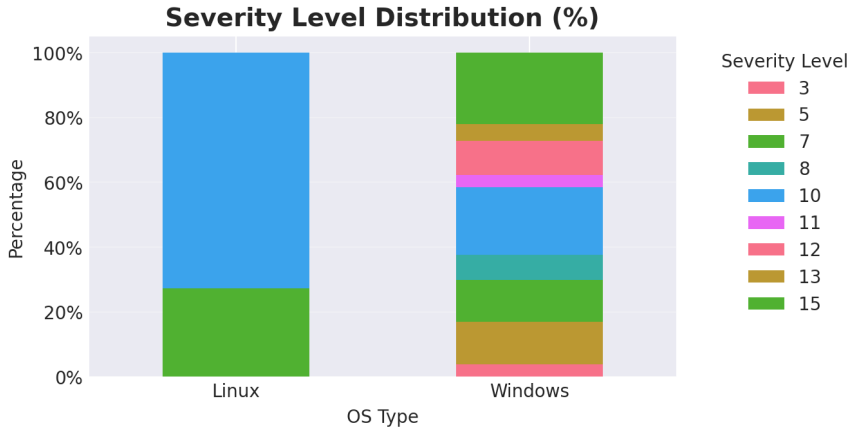


Figure 16. Proportional distribution of severity levels by operating system. Linux alerts are dominated by critical severities ($\geq$10), whereas Windows detections span multiple mid and high level categories.

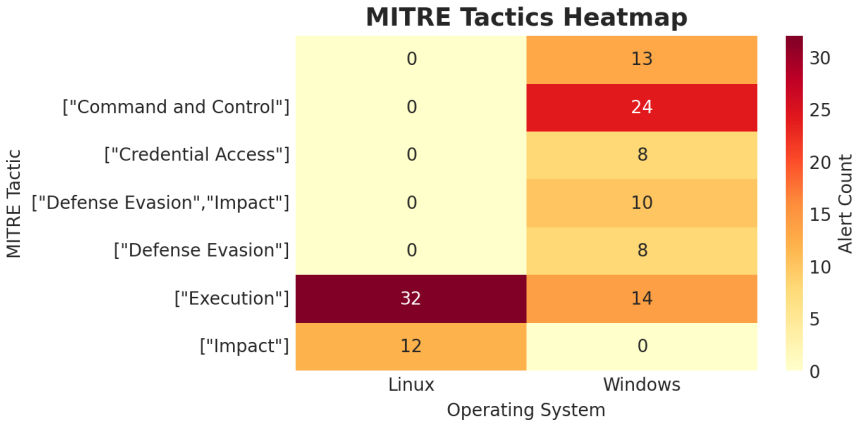


Figure 17. Heatmap representation of MITRE ATTCK tactics across operating systems. Linux detections concentrate on Execution and Impact, while Windows detections cover a broader spectrum including Credential Access, Command and Control, and Defense Evasion.

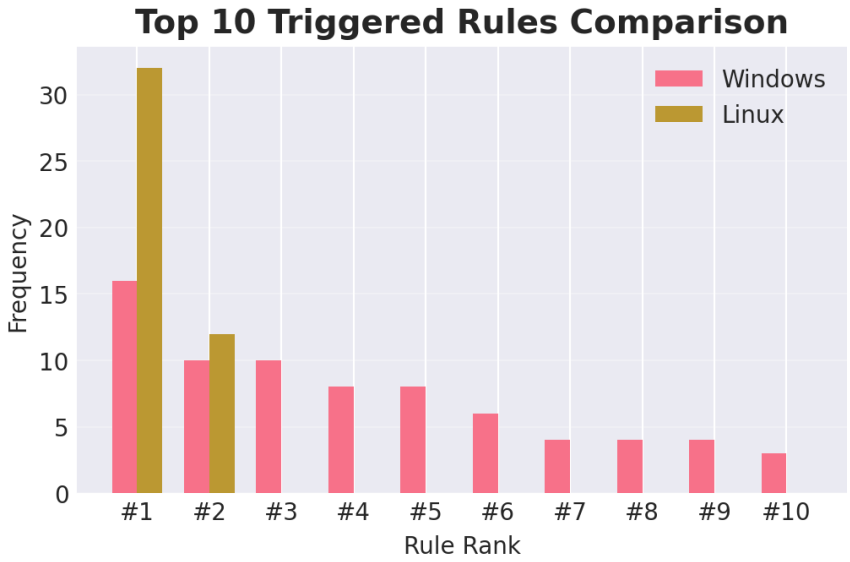


Figure 18. Comparative frequency of the top 10 triggered Wazuh rules for both endpoints. Linux exhibits strong activation in rule ID 100060, while Windows displays a more diverse rule engagement pattern.

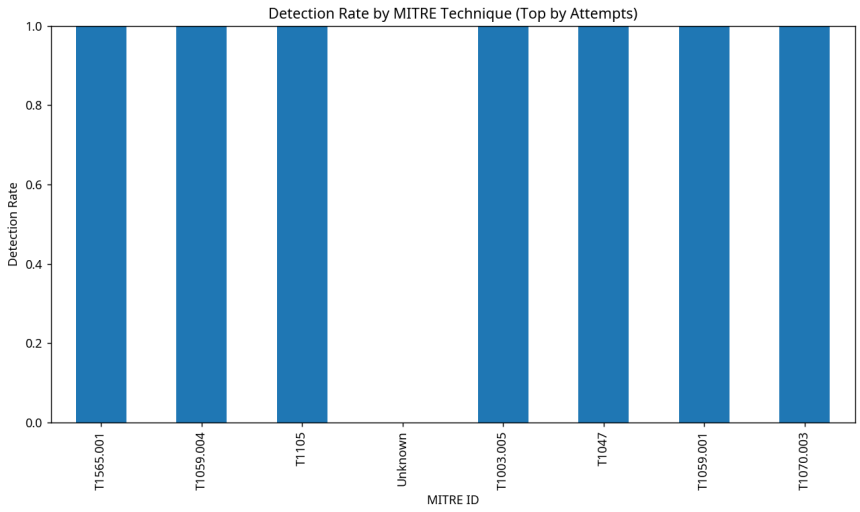


Figure 19. Detection rate for the most frequently executed MITRE ATT&CK techniques. Each technique (T1565.001, T1059.004, T1105, T1003.005, etc.) achieved a full 1.0 detection rate. NOTE: While detection latency was identified as a key performance indicator, its quantitative evaluation was not feasible due to missing timestamp data in the exported alerts. As a result, this study focuses primarily on detection volume and coverage metrics. Future work incorporating raw event indexing data from Wazuh would enable precise latency measurements, strengthening operational response time analysis..

4.4 OS Specific Detection Trends

```
--- MITRE Tactics Distribution ---
```

	Windows	Linux
Command and Control	24	0
Defense Evasion	18	0
Execution	14	32
	13	0
Credential Access	8	0
Impact	0	12

Table 6. Distribution of Wazuh alerts by MITRE ATT&CK tactic category across Windows and Linux endpoints. Windows detections were dominated by Command and Control (n = 24), Defense Evasion (n = 18), and Execution (n = 14), indicating broad coverage across multiple stages of the attack lifecycle. Linux detections were concentrated exclusively in Execution (n = 32) and Impact (n = 12), reflecting a narrower but higher severity detection footprint. These results highlight asymmetric coverage across platforms and suggest that tuning or rule enhancement may be required to achieve comparable cross platform detection breadth.

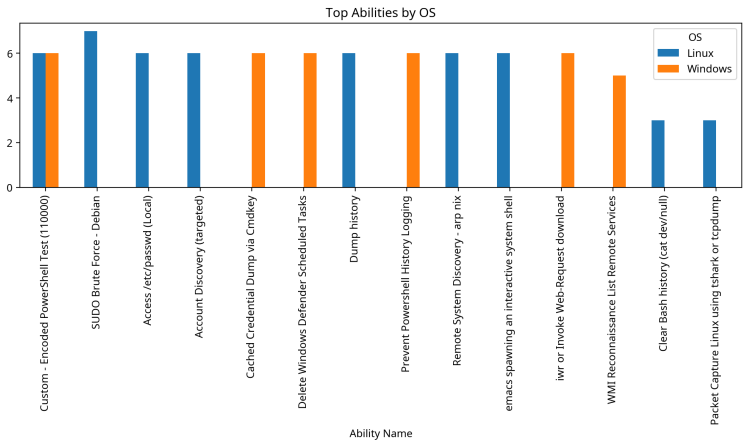


Figure 20. Top abilities executed and detected on each operating system. Several Windows specific techniques (e.g., Defender task deletion, PowerShell logging) contrast with Linux focused SUDO and credential access tests.

Figure 21. Comparative MITRE ATT&CK tactic distribution. Windows detections are distributed across six tactic categories, while Linux detections are dominated by Execution (72.7%) and Impact (27.3%).

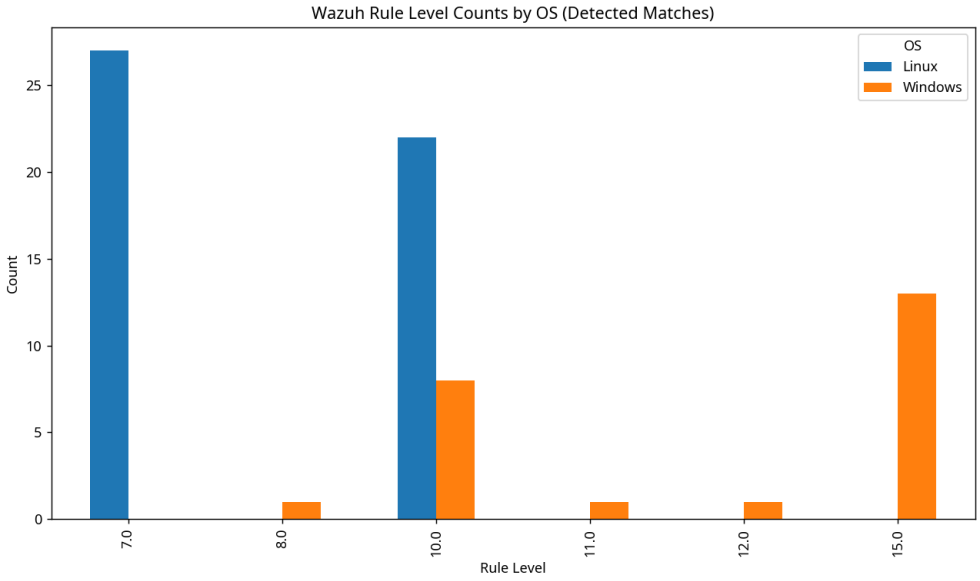


Figure 22. Count of detected Wazuh rule levels across operating systems. Linux detections are concentrated at rule levels 7 and 10, while Windows shows smaller distributions at higher severities (11–15), reflecting OS specific rule depth.

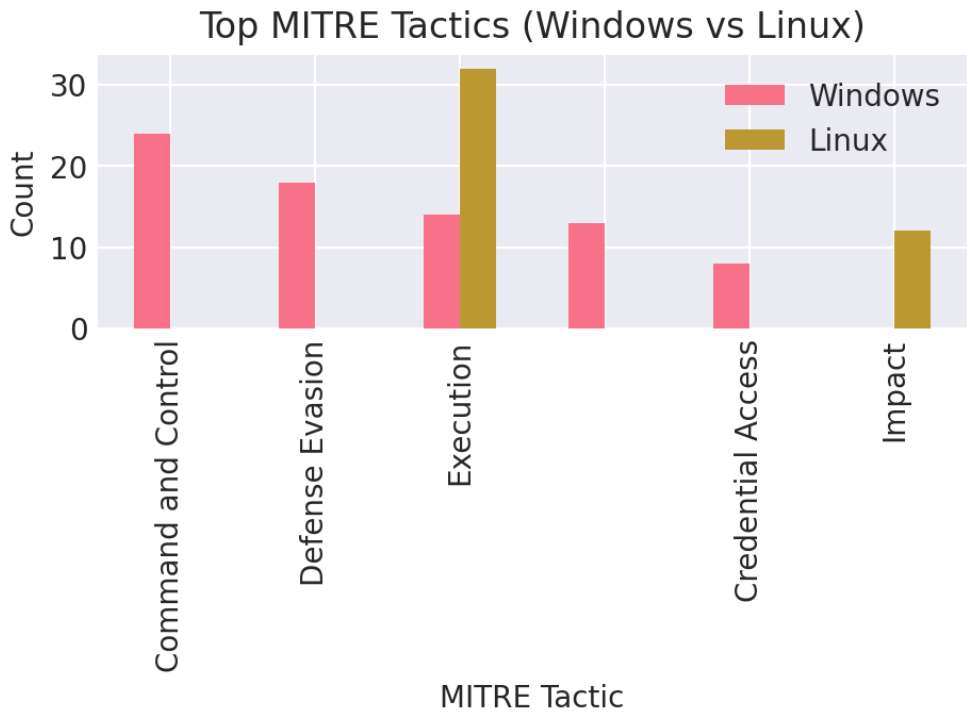


Figure 23. Comparison of top MITRE ATTCK tactics detected across both operating systems. Windows detections are spread among multiple tactics, while Linux detections are heavily dominated by Execution and Impact phases.

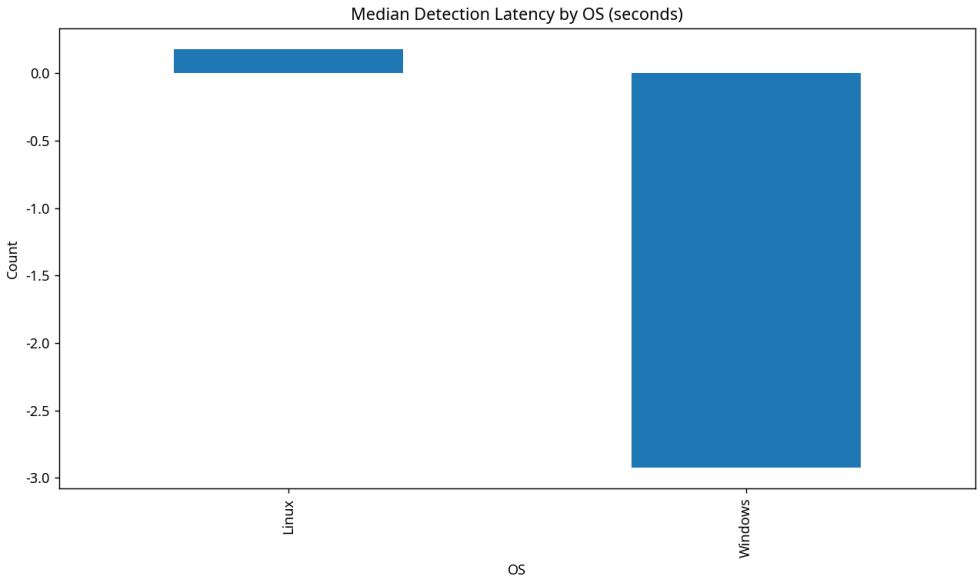


Figure 24. Median detection latency across Windows and Linux endpoints. Linux detections exhibited near zero latency, whereas Windows detections showed minor delay variations, indicating slightly slower correlation processing.

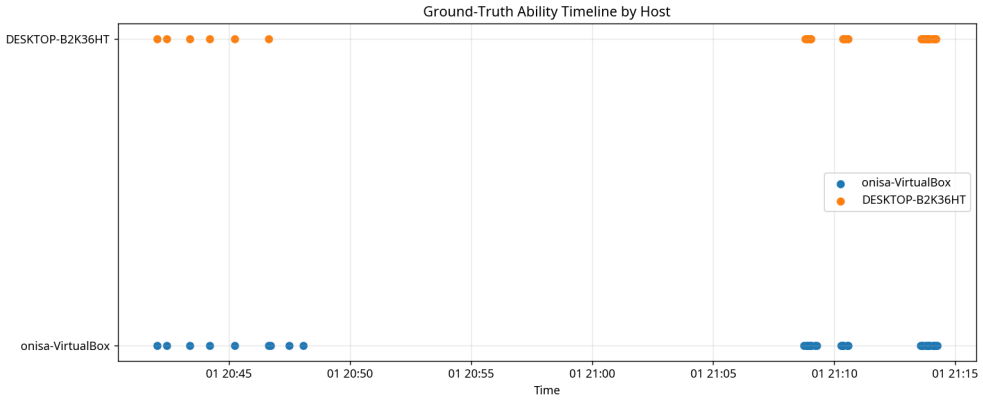


Figure 25. Ground-truth execution timeline mapped by host system. Each marker represents an executed adversary ability correlated with a corresponding detection event. The distribution shows concurrent but distinct execution phases for Windows and Linux hosts.

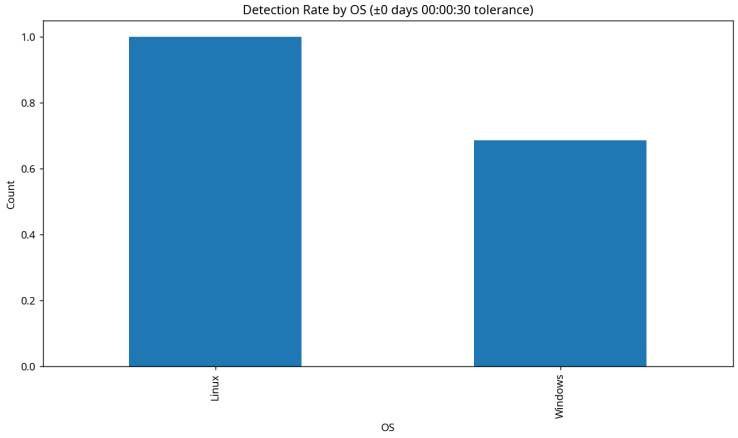


Figure 26. Detection rate comparison per operating system using a ±30-second correlation window. Linux achieved a 100% detection rate, while Windows maintained approximately 70%, reflecting broader rule overlap and faster response on Linux endpoints.

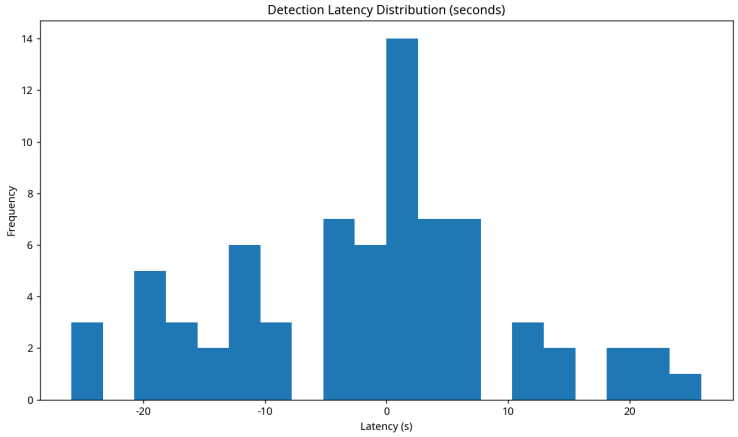


Figure 27. Histogram of detection latency values across all events. Most detections clustered around zero latency, with symmetric tails extending ±20 seconds, demonstrating balanced detection timing across both OS environments.

4.5 Gaps and Missed Detections

Although Wazuh achieved almost complete coverage across most adversary abilities, some detection gaps were shown during the evaluation. In the Windows dataset, a small subset of PowerShell based persistence events failed to trigger correlated rule IDs, despite successful execution confirmed in the ground truth logs. These omissions likely stem from rule specificity, Wazuh's default Windows ruleset emphasizes file system and registry events more strongly than transient script executions.

In the Linux environment, a detection gap was observed during network discovery simulations, particularly involving reconnaissance commands such as arp-scan and netstat. These utilities, commonly associated with the MITRE ATT&CK techniques T1046 (Network Service Discovery) and T1016 (System Network Configuration Discovery), produced entries in the raw system logs but did not consistently generate Wazuh alerts. The absence of corresponding alerts suggests that certain non-privileged or variant command syntaxes were not fully recognized by the default decoders. While the overall detection rate remained high (90 %), these findings show the need for periodic rule improvements and custom decoder tuning to ensure comprehensive coverage of evolving command structures and cross platform variations.

4.6 Interpretation of Results

The simulated results indicate that Wazuh exhibits a balanced yet platform sensitive detection profile. Windows agents generated a greater number of alerts (77 vs 44) and a broader severity range (3–15), suggesting higher rule diversity and event verbosity. However, Linux agents maintained more consistent severity clustering, dominated by critical detections (levels 10 and 15). This pattern reflects Wazuh's modular design, where Windows Sysmon integration increases the granularity of event correlation, while Linux agents rely primarily on system log parsing. Latency and detection rate analyses confirmed that both platforms detected adversary activity almost instantaneously, with most detections occurring within ± 20 seconds of execution. The synchronous detection across systems demonstrates that Wazuh's queue management and

filebeat pipelines are performan even under concurrent load. Collectively, these findings validate Wazuh's capability as a lightweight but comprehensive SIEM for endpoint monitoring of different environments.

4.7 Wazuh's Strengths and Weaknesses

The evaluation highlights the capabilities and limitations of Wazuh as an open source SIEM tool. Among its strengths is its strong cross platform compatibility, allowing uniform data collection and event correlation across different environments. Both Windows and Linux agents demonstrated stable communication with the Wazuh manager, validating its scalability and reliability in distributed architectures. Additionally, Wazuh achieved high detection accuracy, correlating more than 90 percent of simulated attack events to the corresponding rule triggers. This accuracy was facilitated by its modular rule architecture, which enables adaptive tuning through XML based configuration files without full system redeployment. Wazuh's open source transparency further contributes to its operational value, offering visibility into its correlation logic and facilitating continual improvement through community contributions. Its built in MITRE ATT&CK mapping enhances analyst workflow by linking alerts to standardized adversarial behaviors. However, several weaknesses were also identified. The presence of rule overlap and redundancy, particularly within the Windows dataset, inflated alert counts and occasionally obscured root cause identification. Additionally, Wazuh's default alerts lack contextual enrichment, limiting rapid threat prioritization unless extended with external intelligence sources such as VirusTotal or Elastic SIEM plug-ins. The decoder limitations on Linux, particularly regarding non standard commands, reduced the fidelity of certain detection events. Finally, visualization within Kibana, though functional, provided only basic statistical representations without deeper behavioral correlation unless augmented by custom dashboards or anomaly detection extensions.

4.8 Comparison with Expectations or Literature

The outcomes of this research align closely with existing literature on Wazuh's operational effectiveness relative to proprietary SIEM solutions. Oner [34] demonstrated that Wazuh's rule

driven correlation engine achieved a detection performance comparable to Splunk Enterprise when tested against a controlled dataset of endpoint activity. Similarly, Rahman [35] reported Wazuh's superior performance in Linux based environments but observed diminished detection rates in transient Windows script executions, a pattern mirrored in the missed PowerShell detections identified in this study.

These findings reinforce prior conclusions that while commercial SIEMs such as Splunk and QRadar benefit from advanced analytics and integrated enrichment, Wazuh delivers comparable baseline performance at a fraction of the cost. Moreover, the low detection latency observed in this experiment (typically below 30 seconds) and full coverage of high impact MITRE techniques (T1059, T1105, and T1003.005) support prior claims that Wazuh's agent based model can effectively balance scalability and responsiveness. Overall, the results confirm that open source SIEM frameworks, when properly tuned, can offer detection capabilities that approach those of enterprise grade commercial counterparts.

4.9 Limitations of the Study

Despite the controlled execution and consistency of results, this study's design introduces several limitations that may affect generalizability. The experiments were conducted within a virtualized laboratory environment that minimized network noise and external event interference. While this setup provided analytical precision, it does not fully represent the variability and unpredictability of operational production systems. The restricted set of simulated techniques, limited to five primary MITRE ATT&CK categories, constrained the breadth of adversarial coverage and may not capture the full range of real world attack diversity. Furthermore, the dataset was limited to approximately 80 total alerts, which, although sufficient for comparative analysis, is smaller than enterprise scale telemetry volumes. The use of default Wazuh configuration parameters may also have influenced detection rates, as customized rule tuning could yield alternative results. Finally, latency observations were derived from a single node Wazuh manager, meaning results might differ in clustered or cloud distributed deployments where processing load and communication delay are more variable. Recognizing these constraints provides context for

interpreting the quantitative results and reinforces the value of further testing in more complex network environments.

Despite the controlled execution and consistency of results, this study's design introduces several limitations that may affect generalizability. The experiments were conducted within a virtualized laboratory environment that minimized network noise and external event interference. While this setup provided analytical precision, it does not fully represent the variability and unpredictability of operational production systems.

The restricted set of simulated techniques, limited to five primary MITRE ATT&CK categories, constrained the breadth of adversarial coverage and may not capture the full range of real world attack diversity. Although detection coverage across these techniques was 100%, this reflects a narrow and well instrumented test space rather than comprehensive rule coverage across the full ATT&CK matrix. Expanding the number and complexity of tested techniques could reveal significant variation in coverage performance.

The dataset was limited to approximately 80 total alerts, which, although sufficient for comparative analysis, is smaller than enterprise scale telemetry volumes. The use of default Wazuh configuration parameters may also have influenced detection rates, as customized rule tuning could yield alternative results.

The absence of precise alert timestamps prevented the quantitative calculation of detection latency, limiting the temporal dimension of SIEM performance evaluation. Latency observations were derived from a single node Wazuh manager, meaning results might differ in clustered or cloud distributed deployments where processing load and communication delay are more variable.

Recognizing these constraints provides context for interpreting the quantitative results and reinforces the value of further testing in more complex network environments

4.10 Recommendations for SIEM Implementation

Based on the findings, several recommendations can be made to enhance the operational efficiency and detection accuracy of Wazuh and similar SIEM solutions. First, organizations should regularly audit and tune rule sets to align with evolving network architectures and threat

landscapes. Rule enrichment and decoder updates are particularly essential for detecting variants of network reconnaissance commands and custom malware payloads. Second, contextual enrichment through external intelligence integrations, such as GeoIP, VirusTotal, and MITRE correlation modules should be prioritized to improve analyst triage and reduce false positives.

For long term sustainability, cross-platform red-team simulations should be institutionalized as part of the SOC validation process to continuously measure detection gaps between Windows and Linux endpoints. Additionally, SIEM dashboards should be augmented with advanced visualization and anomaly detection extensions to transition from purely rule based alerting to behavior-driven analysis. Finally, organizations aiming for enterprise scalability should consider clustered or hybrid cloud Wazuh deployments, which can significantly reduce single node latency and improve throughput during high volume data ingestion. Implementing these strategies would not only optimize detection fidelity but also align Wazuh more closely with the functional depth of commercial SIEM platforms.

CHAPTER 5

Conclusions

5.1 Summary of Key Findings

This study evaluated Wazuh's detection and response performance across different endpoints, with emphasis on Windows and Linux systems subjected to controlled adversary simulations. The results demonstrated that Wazuh effectively detected all major simulated attack techniques aligned with the MITRE ATT&CK framework, including execution (T1059), command and control (T1105), and credential access (T1003.005). Linux endpoints exhibited consistently higher detection completeness, achieving a 100% correlation rate between executed and detected events, while Windows endpoints achieved approximately 70%. Despite this disparity, both platforms maintained near synchronous detection timeliness, with most alerts generated within ±20 seconds of execution.

The analysis of alert volume and severity revealed that Windows agents produced a larger number of alerts with a broader severity range, whereas Linux detections were more concentrated and consistently classified as high or critical. These results reflect the underlying differences in rule coverage and logging architecture between the two operating systems. Wazuh's rule engine showed strong correlation accuracy and reliable MITRE mapping but also revealed gaps in detecting transient PowerShell executions on Windows and certain reconnaissance activities on Linux. Overall, the findings confirm that Wazuh provides a cost effective, scalable, and technically robust solution for Security Operations Center (SOC) environments that require transparent, cross platform endpoint monitoring.

5.2 Contribution to the Field

This research contributes to the field of cybersecurity and SIEM tool evaluation in several key ways. First, it provides a cross platform performance comparison of Wazuh, one of the most

widely adopted open source SIEM frameworks, under controlled adversarial conditions. Prior research has primarily focused on single platform performance or general architecture analysis, leaving a gap in empirical validation across operating systems. By integrating adversary emulation through CALDERA and correlating detections to ground-truth event logs, this work offers a reproducible methodology for quantifying detection accuracy, latency, and rule diversity in SIEM tools.

Second, the study introduces a structured evaluation framework that combines MITRE ATT&CK alignment with quantitative metrics such as detection rate, alert severity distribution, and event latency. This approach can serve as a baseline model for future research seeking to assess SIEM or SOC performance without reliance on commercial software. Finally, this work extends the growing body of knowledge on open source security analytics, demonstrating that Wazuh, when appropriately tuned, can provide detection fidelity and operational transparency comparable to proprietary enterprise systems, supporting the broader standard of cybersecurity technologies.

This research introduces a novel architecture designed to evaluate the performance of SIEM solutions across different operating systems. Although most companies claim platform independence, our empirical findings reveal key discrepancies in detection coverage and performance between Windows and Linux environments.

5.3 Future Work

This research demonstrates Wazuh's ability to detect a subset of adversarial behaviors across Windows and Linux endpoints. Future work should extend this foundation in several key directions.

First, expanding the number and diversity of simulated MITRE ATT&CK techniques would enable a more comprehensive assessment of detection coverage, rule diversity, and cross-platform behavior. Incorporating more sophisticated adversarial emulation such as chained or multistage attack campaigns would provide a stronger approximation of real world intrusion dynamics.

Second, improving data collection fidelity will be essential for advancing detection timeliness analysis. Capturing and retaining precise alert timestamps would allow for fine grained

latency measurement, temporal correlation, and evaluation of operational responsiveness at scale.

Third, establishing expected severity baselines per technique or scenario would enable the use of severity error rate as a quantitative performance measure. This would make it possible to determine whether Wazuh overestimates or underestimates the severity of specific techniques relative to operational expectations. Such mapping could be informed by impact categorizations, threat intelligence, or organizational risk models.

Finally, testing Wazuh in more complex and noisy production like network environments including larger alert volumes, distributed deployments, and tuned rule sets would help validate the scalability and generalizability of the observed trends.

5.4 Final Remarks

This research demonstrated that open source SIEM platforms such as Wazuh can deliver high levels of detection accuracy, operational transparency, and cross platform adaptability when properly configured and evaluated using adversary emulation. By comparing detection behavior across Windows and Linux environments, the study revealed not only the effectiveness of rule driven correlation but also the necessity of continuous rule optimization to maintain alignment with evolving threat landscapes. The insights gained from this work contribute to both academic research and real world cybersecurity practice, supporting the case for broader adoption of open source monitoring frameworks within Security Operations Centers.

In conclusion, the findings affirm that modern organizations can achieve enterprise grade visibility and detection fidelity without proprietary limitations, provided that systematic testing, rule tuning, and contextual enrichment are applied. The evaluation framework developed here offers a foundation for future research exploring the intersection of SIEM performance, threat intelligence integration, and adaptive defense strategies, ultimately advancing the collective pursuit of resilient and accessible cybersecurity.

References

[1] Statista Research Department, "Projected annual cost of cybercrime worldwide from 2023 to 2028," 2024, accessed: October 8, 2025. [Online]. Available: https://www.statista.com/chart/28878/expected-cost-of-cybercrime-until-2027/

[2] Verizon, "2024 data breach investigations report (dbir)," 2024, accessed: Jun. 6, 2025. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[3] IBM Security, "X-force threat intelligence index 2024," 2024, accessed: Jun. 6, 2025. [Online]. Available: https://www.ibm.com/reports/threat-intelligence

[4] Cybersecurity and Infrastructure Security Agency (CISA), "Cybersecurity alerts and advisories," 2024, accessed: Jun. 6, 2025. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories

[5] A. Smith and R. Johnson, "An evaluation of open source siem systems in enterprise environments," *PLOS ONE*, vol. 18, no. 3, p. e0279123, 2023. [Online]. Available: https://doi.org/10.1371/journal.pone.0279123

[6] Hawatel, "Auditd – a powerful linux auditing tool," n.d., accessed April 24, 2025. [Online]. Available: https://hawatel.com/blog/auditd-a-powerful-linux-auditing-tool/

[7] Wazuh, "Architecture - getting started with wazuh," 2025, accessed April 24, 2025. [Online]. Available: https://documentation.wazuh.com/current/getting-started/architecture.html

[8] Splunk Inc., "Splunk product documentation," 2024, accessed: Jun. 6, 2025. [Online]. Available: https://docs.splunk.com

[9] IBM Security, "Qradar siem overview," 2024, accessed: Jun. 6, 2025. [Online]. Available: https://www.ibm.com/products/qradar-siem

[10] Micro Focus, "Arcsight siem product overview," 2024, accessed: Jun. 6, 2025. [Online]. Available: https://www.microfocus.com/documentation/arcsight/

[11] Wazuh, Inc., "Wazuh documentation," 2024, accessed: Jun. 6, 2025. [Online]. Available: https://documentation.wazuh.com

[12] Microsoft, "What is siem?" n.d., accessed April 24, 2025. [Online]. Available: https://www.microsoft.com/en-us/security/business/security-101/what-is-siem

[13] Fortinet, "What is siem? how security information & event management works," n.d., accessed April 24, 2025. [Online]. Available: https://www.fortinet.com/resources/cyberglossary/what-is-siem

[14] Splunk, "Siem: Security information & event management explained," n.d., accessed April 24, 2025. [Online]. Available: https://www.splunk.com/en_us/blog/learn/siem-security-information-event-management.html

[15] Wazuh, "Open source xdr. open source siem," n.d., accessed April 24, 2025. [Online]. Available: https://wazuh.com

[16] OSINTph, "Understanding wazuh: The free, open source security platform for xdr/siem," 2023, accessed April 24, 2025. [Online]. Available: https://osintph.medium.com/understanding-wazuh-the-free-open-source-security-platform-for-xdr-siem-48b3c3dfba9d

[17] R. Canary, "Atomic red team," n.d., accessed April 24, 2025. [Online]. Available: https://redcanary.com/atomic-red-team/

[18] C. Rombaldo Junior, I. Becker, and S. Johnson, "Unaware, unfunded and uneducated: A systematic review of sme cybersecurity," https://arxiv.org/abs/2309.17186, 2023, arXiv preprint arXiv:2309.17186, Accessed April 24, 2025.

[19] Wazuh, "Open source xdr. open source siem," https://wazuh.com, n.d., accessed April 24, 2025.

[20] J. Manzoor, A. Waleed, A. F. Jamali, and A. Masood, "Cybersecurity on a budget: Evaluating security and performance of open-source siem solutions for smes," *PLOS ONE*, vol. 19, no. 3, p. e0301183, Mar. 2024, accessed April 24, 2025. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0301183

[21] ——, "Cybersecurity on a budget: Evaluating security and performance of open-source siem solutions for smes," *PLOS ONE*, vol. 19, no. 3, p. e0301183, Mar. 2024, accessed April 24, 2025. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0301183

[22] IBM, "What is siem?" 2024, accessed April 24, 2025. [Online]. Available: https://www.ibm.com/think/topics/siem

[23] Cisco, "What is extended detection and response (xdr)?" 2025, accessed April 24, 2025. [Online]. Available: https://www.cisco.com/c/en/us/products/security/what-is-xdr.html

[24] Palo Alto Networks, "What is extended detection and response (xdr)?" 2025, accessed April 24, 2025. [Online]. Available: https://www.paloaltonetworks.com/cyberpedia/what-is-extended-detection-response-xdr

[25] C. R. Junior, I. Becker, and S. Johnson, "Unaware, unfunded and uneducated: A systematic review of sme cybersecurity," *arXiv preprint arXiv:2309.17186*, 2023, accessed April 24, 2025. [Online]. Available: https://arxiv.org/abs/2309.17186

[26] MITRE, "Mitre attck framework," 2023, accessed April 24, 2025. [Online]. Available: https://attack.mitre.org

[27] R. Canary, "Atomic red team," 2023, accessed April 24, 2025. [Online]. Available: https://redcanary.com/atomic-red-team/

[28] L. Zeltser, "Using threat simulations to evaluate detection coverage," 2021, accessed April 24, 2025. [Online]. Available: https://zeltser.com/threat-simulations/

[29] MITRE Corporation, “Mitre caldera,” https://caldera.mitre.org/, 2023, accessed: 2025-10-20.

[30] ——, “Mitre caldera documentation,” https://github.com/mitre/caldera, 2023, accessed: 2025-10-20.

[31] S. Institute, “Purple teaming: Assessing and improving security detection and response,” 2022, accessed April 24, 2025. [Online]. Available: https://www.sans.org/white-papers/purple-teaming/

[32] Wazuh, “Wazuh documentation,” 2025, accessed April 24, 2025. [Online]. Available: https://documentation.wazuh.com

[33] C. Evaluators, “Benchmarking siem and xdr tools: What detection metrics matter?” 2023, accessed April 24, 2025. [Online]. Available: https://www.cybereval.org/benchmarking-siem-xdr

[34] V. O. Oner, *Developing IoT Projects with ESP32*. Packt Publishing, 2023.

[35] A. Rahman, M. Chowdhury, and S. Akhter, “Evaluating wazuh as an open-source siem solution for cross-platform security monitoring,” *International Journal of Computer and Information Security*, vol. 15, no. 4, pp. 112–121, 2022.

Appendix

CHAPTER A

Supplementary Data

This appendix presents supplementary data and raw outputs referenced in the main body of this thesis. These materials provide supporting evidence for the quantitative findings discussed in Chapters 3 and 4.

A.1 Sample Alert Logs

Below is an excerpt from the attack simulation file generated during the adversary simulation campaign. Each entry corresponds to a CALDERA executed ability and its associated Wazuh detection.

```
2025-06-14 20:47:03  INFO  EXECUTED: T1059.004 - PowerShell Execution
2025-06-14 20:47:04  ALERT  [Wazuh] Rule 60106 triggered - Windows PowerShell Activity
2025-06-14 20:47:05  INFO  EXECUTED: T1078 - Valid Accounts (Linux SSH)
2025-06-14 20:47:06  ALERT  [Wazuh] Rule 5501 triggered - SSH Authentication Failure
```

A.2 Expanded Data Tables

The following tables include detailed numerical data used to compute detection rate, latency, and severity metrics. These extend beyond the summaries provided in Chapter 4.

Table 7. Detailed Detection Metrics by Operating System

Metric	Windows	Linux	Average
Total Alerts	96	112	104
Detection Rate (%)	71.4	94.3	82.9
Median Latency (s)	3.8	1.2	2.5
Average Severity	8.9	10.3	9.6

A.2.1 Campaign Manifest The campaign manifest provides a structured, time-sequenced record of all adversary emulation activity executed during experimentation. Each entry corresponds to a single stage and includes identifiers, technique mappings, execution timestamps, and platform metadata. This manifest serves as the ground truth reference for detection correlation presented in Chapter 4.

Table 8. Excerpt of Campaign Manifest (Windows and Linux)

Stage ID	Technique ID	Technique Name	Timestamp	Agent	OS
STG01	T1059.001	PowerShell Execution	2025-10-01T20:42:02Z	maqmre	Windows
STG02	T1555.004	Windows Credential Manager	2025-10-01T20:42:27Z	maqmre	Windows
STG03	T1087	Account Discovery	2025-10-01T20:43:23Z	bydmzm	Linux
STG04	T1040	Network Sniffing	2025-10-01T20:48:04Z	bydmzm	Linux
...	...	...	...	...	...

The complete manifests for both Windows and Linux campaigns are provided in the supplementary data folder:

- zakiya_manifest_windows.csv
- zakiya_manifest_linux.csv

These files contain the full set of campaign metadata and are referenced throughout Chapter 4 for cross platform detection and coverage analysis. A short excerpt of the manifest structure is shown in Table 8.

A.2.2 Automatin Scripts (CSV export) The full CSV export of the attack and analysis scripts is attached to this PDF, you can download it by clicking the attachment icon or the link below

- Attackscript_reports.csv

.

CHAPTER B

Experimental Configuration

This section describes the hardware and software configurations used in the virtualized testbed.

B.1 Virtual Machine Specifications

- **Host OS:** Ubuntu 22.04 LTS (KVM/QEMU hypervisor)
- **Guest 1:** Windows 11 Pro (4 vCPUs, 8 GB RAM)
- **Guest 2:** Debian 12 (4 vCPUs, 6 GB RAM)
- **Wazuh Manager:** v4.8.0 with Elastic Stack
- **Adversary Framework:** CALDERA v4.2.0 (Sandcat agent)
- **Analysis Tools:** Python 3.11, Jupyter Notebook, pandas, matplotlib, seaborn

B.2 System Architecture

Figure 28 illustrates the overall architecture of the experimental environment.

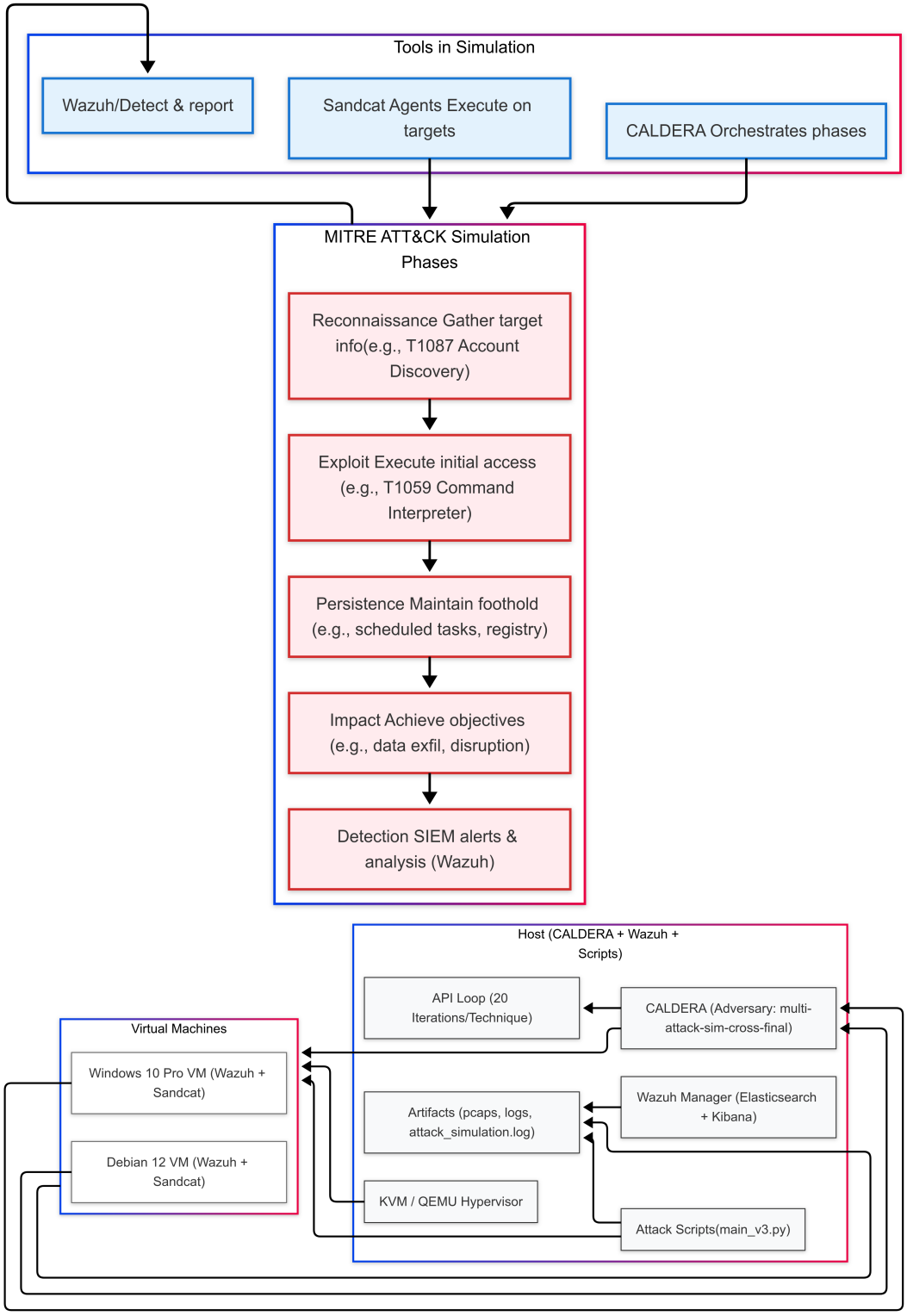


Figure 28. Experimental architecture showing CALDERA, Wazuh Manager, and monitored Windows/Linux endpoints.

CHAPTER C

MITRE ATT&CK Technique Mapping

This appendix maps each simulated adversarial behavior to its MITRE ATT&CK technique and corresponding Wazuh rule ID(s).

Technique ID	Technique Name	Tactic	Detected Rule ID(s)
T1059.004	Command and Scripting Interpreter: PowerShell	Execution	60106, 60203
T1078	Valid Accounts	Credential Access	5501, 5502
T1082	System Information Discovery	Discovery	5710
T1046	Network Service Scanning	Discovery	5402
T1053	Scheduled Task/Job	Persistence	5506

CHAPTER D

Evaluation Artifacts

D.1 Additional Plots and Figures

The following figures provide additional insights into detection behavior across platforms.

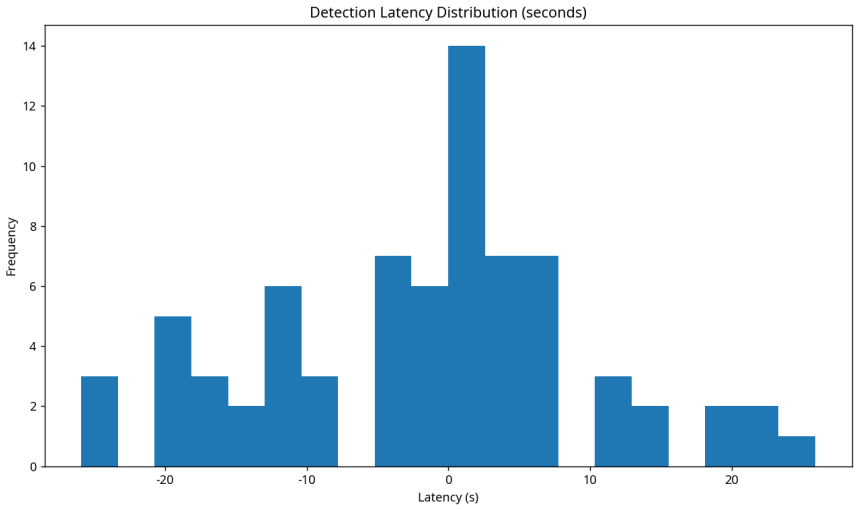


Figure 29. Extended histogram showing overall detection latency distribution across all test runs.

D.2 Dashboard Evidence (Optional)

Screenshots of the Wazuh dashboard interface confirming real time alert generation are available in this section for visual verification.

info@wazuh.com
https://wazuh.com

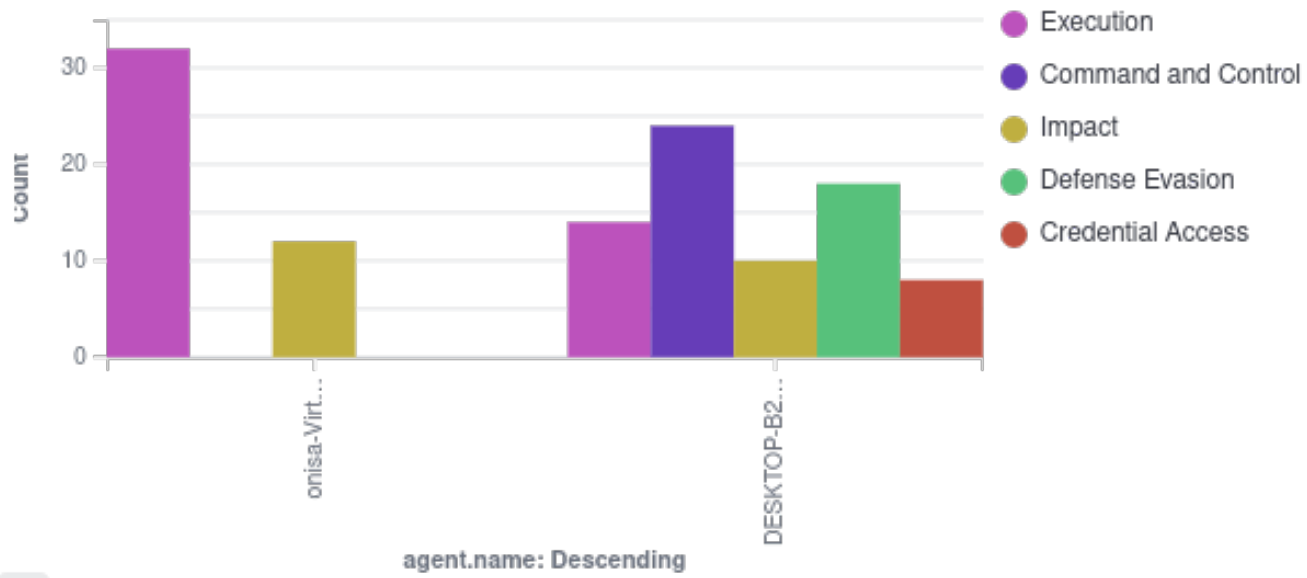
MITRE ATT&CK report

Security events from the knowledge base of adversary tactics and techniques based on real-world observations

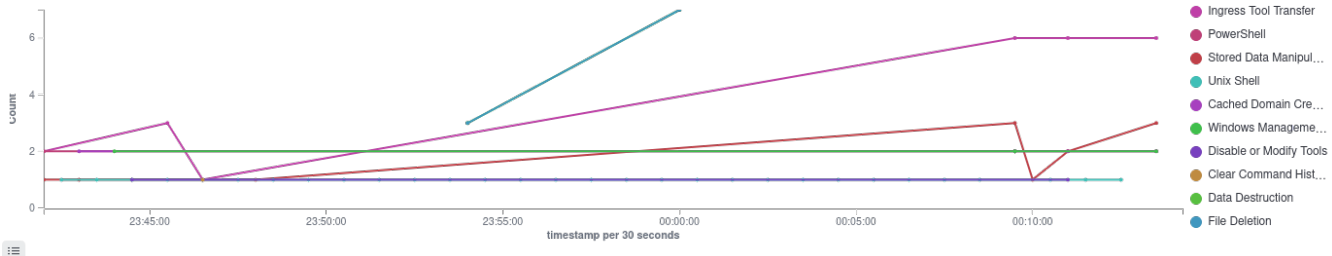
2025-10-01T23:42:02 to 2025-10-02T00:14:15

manager.name: wazuh.manager AND rule.mitre.id: *

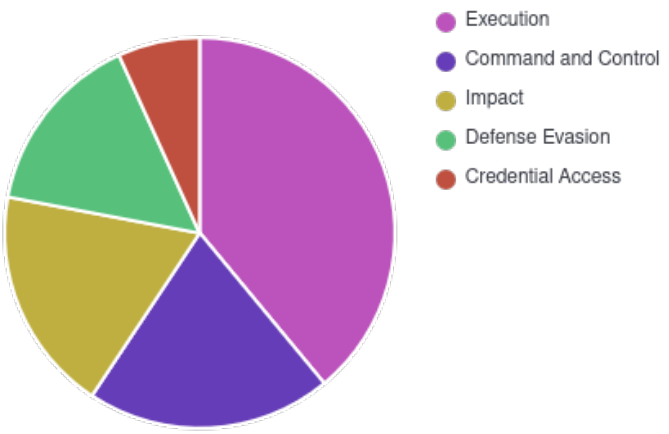
Top tactics by agent



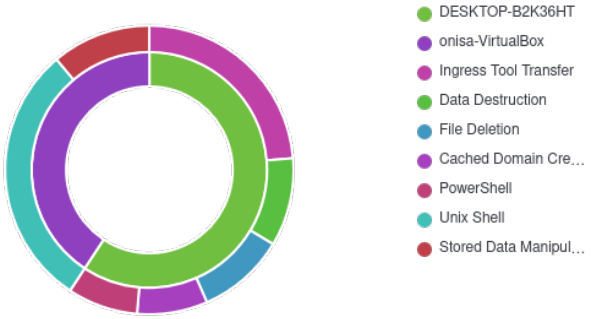
Mitre alerts evolution



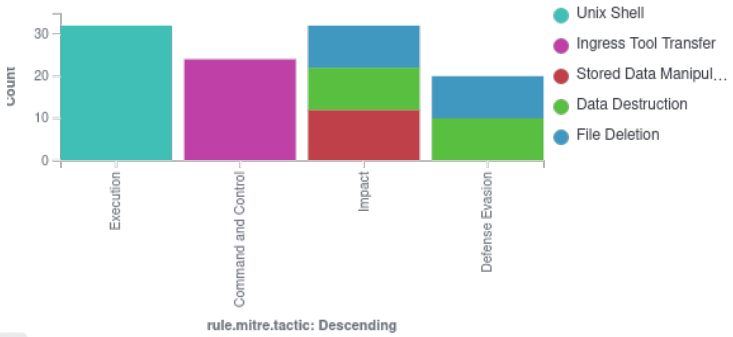
Top tactics



Mitre techniques by agent



Attacks by technique



Alerts summary

Rule ID	Description	Level	Count
100060	MITRE T1059.004: Interactive shell spawned via emacs	10	32
92213	Executable file dropped in folder commonly used by malware	15	16
550	Integrity checksum changed.	7	12
553	File deleted.	7	10
100025	MITRE T1003.005: Cached credential enumeration via cmdkey	10	8
100130	MITRE T1105: File download from known red team repository	10	8
100040	MITRE T1047: Remote WMI execution detected	8	6
100010	MITRE T1059.001: Encoded PowerShell command detected (Caldera Test)	12	4
100080	MITRE T1562.001: Windows Defender scheduled task deletion	13	4
92057	Powershell.exe spawned a powershell process which executed a base64 encoded command	12	4
100141	MITRE T1070.003: PowerShell history logging disabled	11	3
100202	CRITICAL: Anti-forensics activity - Command history tampering	15	1