Final Report
SPR888 - Applied Security Project

Prepared by:

Group 3

Ornan Roberts
Syed Mujahid Hamid Ali
Ananthu Krishna Vadakkeppatt
Yannish Kumar Ballachandher Sreedevi

Abstract

The exponential growth of cyber threats in contemporary digital landscapes has created an urgent demand for sophisticated anomaly detection systems capable of identifying and neutralizing malicious activities in real-time before they compromise critical production environments. Modern companies face an increasingly complex threat ecosystem where traditional security measures prove inadequate against advanced persistent threats, zero-day exploits, and sophisticated attack vectors that continuously evolve to bypass conventional detection mechanisms. This project addresses these challenges through the comprehensive design, implementation, and validation of an integrated real-time anomaly detection system that synergistically combines network intrusion detection capabilities, Security Information and Event Management (SIEM) functionality, and honeypot technology to create robust multi-layered defense architecture.

The proposed solution in this paper leverages a carefully orchestrated combination of open-source security tools to deliver enterprise-grade threat detection and response capabilities. The key functionality of the solution is to utilize pattern recognition algorithms to identify deviations between malicious and benign traffic, isolate attackers through intelligent redirection mechanisms and have policy-driven response mechanisms that minimize human intervention. At its core, the system proposed is a robust multi-layered defense mechanism offering containment, behavioral analysis and creating a controlled environment where attackers can be isolated and studied without impacting production systems.

Table of Contents

Table of Figures

Executive Summary

In an era of digital transformation and increasingly sophisticated and complex cyber threats, most modern organizations have to face a huge number of challenges in securing their information systems. As threat actors use more stealthier and targeted attack techniques, most traditional security solutions such as static firewalls, signature-based antivirus software, and conventional intrusion detection systems (IDS) are proving inadequate. These legacy systems often fail to detect complex malicious behaviors, leaving organizations more exposed to breaches, data loss, operational disruption, and reputational damage.

The cybersecurity landscape today is dominated by advanced persistent threats, insider threats, credential-based attacks, and sophisticated zero-day exploits—all of which operate under the radar of static security controls. The lack of real-time detection and containment capabilities in most of the traditional setups leaves critical infrastructure vulnerable during the window of opportunity that attackers exploit. This gap between threat detection and incident response time represents a fundamental flaw in current security architectures.

This research project addresses this issue by designing, implementing, and evaluating an integrated, anomaly-driven defense system that combines real-time detection, traffic analysis, and automated attacker redirection through honeypot engagement. The core innovation of the system lies in its ability to not only identify malicious behavior as it happens but to respond instantly by redirecting the attacker to a simulated environment, isolating threats, and protecting production assets.

Our proposed solution utilizes and integrates three open-source security technologies:

- **Suricata IDS**, a high-performance intrusion detection engine capable of packet-level analysis using customizable rule sets.
- **Wazuh SIEM**, a centralized security information and event management system used to correlate logs, analyze anomalies, and provide visibility through dashboards.
- **Artillery Honeypot**, a lightweight deception system that mimics production services and engages attackers without risking real assets.

Together, these components create a multi-layered defense system capable of identifying deviations from normal or benign network behavior, classifying threats, and executing predefined response mechanisms in real time.

The primary objectives of this project are as follows:

1. **Design and Implementation**: To architect a scalable and effective anomaly-based detection system capable of recognizing malicious activity within a localized network environment.

2. **Traffic Classification**: To develop intelligent filtering mechanisms that can distinguish between legitimate user behavior and potential threat activity using rule-based detection techniques and traffic baselines.

3. **SIEM Integration**: To deploy Wazuh SIEM for aggregating logs from network sources, classifying attack behaviors based on frameworks like MITRE ATT&CK, and providing real-time security event visualization.

4. **Automated Response**: To implement real-time redirection mechanisms that can isolate suspected attackers by directing them into honeypot environments for monitoring and behavioral analysis.

Unlike conventional IDS/IPS solutions that merely generate alerts requiring manual response, this system emphasizes automated containment. Once an anomaly is detected—such as brute-force login attempts, reconnaissance scans, or command injection—the attacker is redirected to the honeypot where their behavior can be safely logged and analyzed. This redirection is handled programmatically, reducing the response time to milliseconds and eliminating the need for constant human oversight.

During testing, various simulated attack scenarios were executed using tools such as Hydra (for SSH and FTP brute-force attacks), Nmap (for network and port scanning), and Hping3 (for IP spoofing). These simulations closely mimicked real-world tactics and techniques. The system was able to detect each attack type promptly, differentiate them from legitimate user behavior, and redirect the traffic from our production server to the honeypot server accordingly. Meanwhile, Wazuh SIEM provided real-time insights into each event, mapping tactics to MITRE ATT&CK techniques such as:

- **T1110** (Brute Force)
- **T1046** (Network Scanning)
- **T1190** (Exploit Public-Facing Application)
- **T1550.002** (Pass-the-Hash)
- **T1078** (Valid Accounts)

This mapping capability improves both situational awareness and post-incident analysis, allowing for more accurate threat intelligence and defense strategy updates.

One of the most critical aspects of this project was the ability to reduce false positives—alerts triggered by benign actions. By leveraging layered detection through Suricata and Wazuh, and by defining behavioral baselines for normal traffic, the system significantly reduced alert noise and prioritized only threats with high-level priority. This approach preserves operational efficiency while enhancing security.

Moreover, the honeypot component offered not only containment but also **threatened intelligence enrichment**. By engaging attackers in a controlled environment, the system gathered valuable information on the behavior of the threat actor, tools being used, the techniques, and procedures. This information can be used to tune rulesets, create and develop new detection signatures, and modify our blue-team defensive playbooks.

The deployment is fully virtualized, using a firewall with network segmentation and policy enforcement. It was tested within a controlled lab using VMware Workstation and could be easily adapted for real-world environments in both on-premises and cloud settings.

The project offers several strategic benefits:

- Drastically reduced response time through automation.
- Real-time visibility into attack patterns and behaviors.
- Improved classification of attack types using the MITRE ATT&CK framework.
- Isolation of threats before they affect production systems or mission-critical systems.
- Enhanced forensic capabilities via honeypot engagement logs.
- Reduced burden on security analysts through intelligent filtering and better alerts.

This research project demonstrates that real-time threat detection paired with using a honeypot and having an automated redirection is a practical, scalable, and highly effective approach to modern cybersecurity defense. By integrating SIEM capabilities with deception and automated containment, the system goes beyond passive detection to offer an initiative-taking, layered security posture. This approach represents a significant advancement over conventional methods and sets a foundation for further work in automated incident response, machine learning integration, and adaptive cyber defense in dynamic network environments.

Introduction

Modern technology has played a huge role in changing the way we live our lives. It has helped us connect in ways we would never have imagined and provided us with access to massive amounts of information. However, there's a downside to all this as the more connected we become, the more opportunities a cybercriminal gains to attack our systems or steal our data. Even though there are security tools like firewalls, antivirus software, intrusion detection systems, and more, there is a constant change in the trend of attacks that hackers today come up with. Hence, the traditional defenses that primarily relied on looking for known patterns of good and bad will not work anymore, as cybercriminals have become smarter. Attackers have started using techniques that look like normal everyday computer activity, which do not get picked up by conventional security systems.

Hackers today no longer break down the front door anymore. Instead, they may use legitimate software already available on your computer, which is also called living off the land in cybersecurity. They might stay on your system to steal credentials or just wait patiently before eventually striking. These stealthy approaches can easily slip past security measures that most organizations rely on. The real problem lies in the fact that current conventional security systems are terrible at noticing when something unusual happens. By the time most companies know there is something wrong, they are breached, and the damage is already done. What started as a small breach could end up becoming a massive data theft, as there is a delay between when the attack begins and when someone notices it.

This delay could be critical as it remains the difference between rapid containment and full-scale data breaches. Therefore, there is a clear need for the development of adaptive and intelligent anomaly detection solutions to prevent more sophisticated attacks. These systems can spot unusual behavior and respond immediately, as traditional detection systems are not adept at real-time monitoring.

Importance of Real-Time Anomaly Detection and Response

The limitations identified in traditional security approaches necessitate a fundamental shift towards anomaly-based threat detection. Instead of relying on predetermined signatures of threats that are already known, anomaly-based detection systems can identify unusual patterns of behavior that tend to deviate from already established baselines, which could potentially indicate a security compromise. However, detection capabilities alone are insufficient in threat environments. The effectiveness of an anomaly detection system critically depends on the integration with automated response mechanisms that can be initiated to contain attacks upon successful threat identification.

This integration of real-time detection along with an automated response system is key for organizations that are responsible for managing sensitive data and critical operations. Financial institutions, healthcare systems, cloud service providers, and others face heightened risks due to the high value of their data assets as well as potential losses that they could face due to security incidents. For these organizations, a lapse in time or delay measured in minutes or hours may result in substantial financial losses, regulatory penalties, operational disruptions, and long-term reputational damage.

The research presented in this paper addresses these challenges through the development of an integrated security framework that combines anomaly detection with deception technologies, along with automated response capabilities. This proposed system incorporates the use of honeypots as a method of strategic deception within the network infrastructure. It can create controlled environments that have the potential to attract and contain malicious actors while preserving the integrity of the production systems. These honeypots will work in conjunction with Security Information Event Management (SIEM) platforms that can provide centralized correlation and analysis of network behavior across the entire network.

This multi-layered approach creates a comprehensive defense model that is not only effective at detecting anomalous activities but also allows controlled spaces for threat analysis as well as automated containment mechanisms. The integration of these components will enable organizations to move beyond a reactive security posture towards more initiative-taking threat management capabilities.

Review of Related Work and Technical Foundations

Several foundational studies have explored the subjects of honeypots and how they can be used for anomaly detection in network security environments while establishing both the potential and limitations of these tools in contemporary cybersecurity implementations.

Baykara and Das (2015) introduced a novel honeypot-based architecture that was designed to work in tandem with real time intrusion detection and prevention systems (IDPS). This approach was successful in not only improving attack visibility but also reduced false positive cases because of the honeypot integration. However, despite their strengths the manual oversight for incident response limits its use in high-speed attack scenarios where automated containment would be necessary. The authors propose a novel honeypot-based system that integrates real-time intrusion detection and prevention systems through this paper. This method is unique as it actively detects attackers by analyzing their behavior within a controlled environment. The approach combines the use of a honeypot with traditional IDPS techniques that help to reduce the number of false positives generated and, at the same time, increase response times. This hybrid system can offer an initiative-taking defense mechanism that can improve network security while learning from attacker

interactions. Hence, this study shows us the potential of implementing honeypots with real-time monitoring security systems to create better cybersecurity solutions.

Qurbonaliyeva and Abduraxmanova (2024) examined technical as well as psychological methods that are used to attract attackers to interact with the honeypot systems. Their work focused on highlighting the effectiveness of strategic placement, realistic simulation of services and behavioral cues that would instead increase attacker engagement. This study focused on attacker interaction rather than defensive outcomes, but it still offered valuable insights improving the realism of honeypots and valuing the fact that it is an essential component for data collection as well as attacker deception. The goal of the authors of this paper is to examine various techniques that can be used to attract attackers to honeypots. Authors believe that the professional usage of a honeypot requires a combination of technical capability that can be used to mimic realistic system services, along with the ability to psychologically manipulate. The article talks about various variations of honeypots and methods like emulating partial vulnerabilities, replicating production setups, or simulating network traffic to lure the attackers. It also addresses strategies that can be used to optimize efficiency, as well as various legal or ethical challenges faced when deploying a honeypot. Findings from this paper indicate that an effective honeypot perfectly balances technical design while also adapting to behavioral patterns to improve cybersecurity defenses.

Mairh et al. (2011) through the paper introduced a foundational classification of honeypots based on their interaction level – low, medium and high. Furthermore, they also discussed their roles in intrusion detection. While high-interaction honeypots are incredibly useful in capturing attacker behavior, the authors emphasized the significant resources and security challenges that are involved in maintaining them. Moreover, the absence of automation as well as real-time containment mechanisms in their architecture brings up a security gap between the detection capabilities and responsive operations. The paper focuses on providing a detailed survey of the use of honeypots in network security. It outlines the architecture of various defensive mechanisms, examining the role of honeypots and their classification based on interaction levels and deployment strategies. It enlightens the readers on how honeypots serve as efficient tools for studying attacker behavior, detecting intrusions early, and gathering intelligence. The authors also discuss the challenges of deployment, maintenance, and other potential legal concerns.

Sethi and Mathew (2021) based their paper on the most recent advancements in honeypot-based network models with a particular focus on incorporating intelligent decision-making systems. They also explored the application of automation and integration with machine learning which were used to actively monitor and respond to threats. The findings suggested that there is a growing convergence between honeypot design and anomaly detection methodologies. Some of the barriers mentioned in the paper include scalability and system complexity. This paper aims to explain various methods that can be used to implement a honeypot system into a network. By proposing multiple solutions that involve dynamic deployment of honeypots, blockchain-based distributed honeypots, and high-interaction IoT-based honeypots. The best method among all of them remains

the blockchain-based distributed honeypot system, as it is scalable and highly efficient. The study concludes by stating that advanced honeypot systems play a huge role in strengthening overall network defense. This research contributes to the ongoing development of smarter, more resilient cybersecurity infrastructures.

In recent years, the research on honeypots has advanced significantly, where the focus has entirely been on understanding attacker behavior, improving system realism and integration of deception into broader cybersecurity. **Zemene and Avadhani (2015)** through their paper experimented with the deployment of a high-interaction honeypot that was specifically designed to study SSH attach techniques. The authors revealed that high-interaction honeypot systems are valuable in identifying deep behavioral patterns. In this case they captured login attempts and brute-force behaviors, their work provided detailed insights into attacker methodologies and credential guessing techniques. They also noted challenges related to resource demands, system scalability and risk containment. The study presented in this paper demonstrates the deployment of a high-interaction SSH honeypot within a campus network. The goal of the paper remains to study common brute force and dictionary attacks, which are focused on the SSH service. Over the period of 45 days, the honeypot recorded attack attempts that ranged over 116,00 in number from a total of 1,574 unique IPs. The paper is valuable as it helps us to analyze attack behaviors that could help us implement better security practices to protect the SSH service.

Mokube and Adams (2007) laid foundational concepts through their research paper that categorized honeypots based on their level of interaction and intent of deployment. Their paper was one of the first to conceptualize honeypots as a tool that can be used not only as decoys but as tools to profile attackers and early detection systems. They also addressed legal and ethical concerns related to data collection as well as entrapment that remain relevant today too. Even though the technical aspects of their work are limited, the conceptual contributions to the broad spectrum of cybersecurity have defined the landscape for later technical explorations. This paper describes a blockchain-based honeypot system that can be used to prevent distributed denial of service attacks and ransomware. The system uses smart contracts on a private Ethereum blockchain that manage as well as verify interactions made with decoy services. The honeypot hence prevents attackers from bypassing detection mechanisms due to the stability and transparency brought out by the blockchain. Results show that security improved along with real-time detection. The study shows us that integrating a blockchain along with a honeypot greatly improves the reliability and scalability of the network against various types of threats.

Javadpour et al. (2024) presented a comprehensive survey that focused on cyber deception techniques that could be used to increase the effectiveness of honeypots. This paper focused on various types of deception strategies that included dynamic interaction, decoy diversification and AI driven attacker engagement. They explored how these techniques could increase the believability of the honeypot system. One of the key contributions of their study was the proposition of the integration of deception along with adaptive systems that constantly evolved in

response to attacker behavior. However, the authors again pointed out limitations in scalability, automation and the difficulty of creating such realistic environments that can consistently fool advanced adversaries. This paper focuses on presenting a comprehensive study that is centered around cyber deception techniques implemented to enhance the performance of a honeypot. The study looks at various different strategies, such as data obfuscation, fake services, as well as dynamic switch of attack surfaces that are used to lure the attackers and analyze their activity. These techniques play a key role in improving the overall ability of the honeypot to detect various types of sophisticated attacks and, at the same time, gather intelligence on new attack behaviors. The authors emphasize the importance of having an adaptive system that can prevent the ever-evolving wave of cyberattacks. Overall, the study shows us how multiple deception mechanisms paired with a honeypot can improve their performance in various network environments.

Nicomette et al. (2010) through the paper detailed the setup and deployment of a high-interactive honeypot in an operational environment. The study presented practical lessons that were learnt during the deployment, attacker behavior analysis, vulnerabilities related to the infrastructure and monitoring limitations. Various challenges documented by the authors included ensuring system security from unintended compromise and the operational burden of managing such a highly demanding environment. While high-interaction honeypots produce rich data, the authors are of the opinion that they should be managed carefully so that they do not turn out to become a liability at the end. This paper goes over the detailed setup of a high-interaction honeypot that was primarily designed to mimic a real system to gain a better understanding of attacker behavior analysis. The experiments conducted by the authors highlight the complexity as well as resource management demands that had to be addressed to maintain such honeypots. The study was key in uncovering how attackers exploit vulnerabilities and interact with the honeypot environment over an extended period of time. Continuous monitoring, careful isolation, and timely response to avoid damage are some of the key lessons that the paper emphasizes. The authors believe that we should think of balancing the realism of the honeypot with safety to maximize the amount of data that we can collect while minimizing risk. This article is key in providing practical insights to cybersecurity professionals who are looking to implement a high-interaction honeypot in an operational network.

AlZoubi and Alrashdan (2022b) through this paper evaluate deploying a honeypot on a network and assessing its impact on overall system security. The study focuses on analyzing how honeypots help in early threat detection, behavior analysis of attacks, as well as the ability to reduce the exposure of critical assets. Results from the study confirm that honeypots effectively divert malicious traffic away from production systems, thereby improving the security posture. However, the authors highlight the importance of proper placement along with configuration to maximize the effectiveness of honeypots. Experimental data obtained from the tests conducted support the claim that the integration of honeypots reduces attack success rates and enhances the security of the network. The research conducted concludes that honeypots are valuable tools that are crucial in building layered cybersecurity strategies.

Together, these studies build a convincing argument to reinforce the potential of honeypot-based systems to serve as both initiative-taking and reactive security tools. However, most of these papers also identify a common shortcoming that is the lack of automation and real time responsiveness. Bridging this gap would be critical in transitioning honeypots from bare observational components to an integral part of the dynamic cyber defense infrastructure.

Project Contribution

This research identifies security gaps through the development of a comprehensive security framework that is integrated with various essential components to provide real-time threat detection, behavioral analysis, as well as automated threat redirection. The proposed system architecture incorporates these three components that work synergistically to provide enhanced security capabilities:

1. The system employs Suricata as an intrusion detection system engine, where we leverage its high-performance packet analysis capabilities to conduct real-time network traffic inspection by implementing rule-based anomaly detection mechanisms. This serves as the foundational layer that can identify potential security incidents at the network level.
2. Secondly, we have implemented Wazuh SIEM that functions as a correlation platform, which is used to aggregate security events from multiple sources that provide us with a comprehensive threat classification capability. The SIEM component also utilizes established frameworks such as MITRE ATT&CK to categorize as well as analyze threat patterns, while offering us visualization options that allow us to identify trends across the security landscape.
3. The third is Artillery Honeypot which is a lightweight deception platform that simulates commonly targeted services such as SSH, FTP and HTTP protocols. This component serves dual purposes: attracting malicious actors away from production systems and at the same time creating controlled environments for behavioral analysis and threat intelligence gathering.

The integration of these components enables the system to transcend traditional detection methodologies that primarily relied on passive monitoring by implementing automated threat redirection capabilities. When malicious activity is identified, the system is trained to automatically channel suspicious traffic towards the honeypot environment, which effectively isolates threats from critical infrastructure while maintaining operational safety.

Methodology

Research Design

The investigation carried out by this paper primarily employs a mixed-methods experimental design, which integrates quantitative performance metrics along with qualitative behavioral analysis to gauge the efficacy of an integrated anomaly-driven honeypot architecture. The research is conducted in a monitored environment along with the systematic evaluation of the proposed system's effectiveness across different threat scenarios while ensuring reproducibility and validity.

The experiment framework is focused on a comparative analysis methodology that measures system performance in different stages – pre-implementation and post-implementation. This design allows us to precisely measure the quantification of performance improvements while also keeping in mind the variables that could affect the validity. The research adopts an iterative method of development, where there is a continuous optimization of detection techniques and response procedures that are based on a set of empirical observations of system behavior and emerging attack vectors.

Research Approach

The methodology adopts a pragmatic research philosophy, focusing on practical solutions to real-world cybersecurity challenges. The approach emphasizes empirical validation through controlled experimentation, enabling quantitative measurement of system effectiveness while maintaining the flexibility to incorporate qualitative insights from threat behavior analysis.

The research design incorporates elements of action research, where the implementation process itself generates insights that inform subsequent iterations of the system. This approach is particularly relevant for cybersecurity research, where threat landscapes continuously evolve and require adaptive defense mechanisms.

Honeypot Role

To ensure that the malicious actors believe that they are attacking a production server while interacting with the honeypot system, we have implemented several deception mechanisms that blur the distinction between a legitimate and a decoy system:

1. Service Simulation: Artillery honeypot deploys fake services that include FTP, SSH, HTTP, and other common protocols that replicate the behavior as well as responses of genuine production servers. This helps us create the illusion of a vulnerable target system that appears to be legitimate when attackers try to conduct reconnaissance or port scanning activities.
2. Network Integration: The honeypot is positioned within the same network range as the production systems, which makes it appear as part of the internal infrastructure. This placement convinces the attackers that they may have successfully penetrated the network perimeter and are operating within a trusted internal zone that contains critical servers.

Production System V/s Honeypot Differentiation

The fundamental difference between production systems and honeypots lies in their operational purpose and data sensitivity:

Production Systems: It contains genuine operational data, supports user operations and maintains critical organizational functions. These systems store sensitive information, process legitimate transactions, and require continuous availability to support business operations. Any compromise of the production system may result in actual data loss, operational disruption or potential regulatory violations.

Honeypots: They contain no legitimate business data and serve no operations or critical organizational functions. They are designed to attract and contain malicious activity while appearing to be identical to a production system. Honeypot systems deliberately invite attack attempts to observe attack behavior that collects threat intelligence, which aids us in analyzing attack methodologies without risking any actual business assets. Any interaction with the honeypot is considered to be inherently suspicious, as legitimate users will have no reason to access these systems.

Proposed Solution

This research proposes a novel, integrated cybersecurity defense system that addresses the critical gap between traditional intrusion detection and modern threat containment by seamlessly integrating anomaly detection, honeypot technology, and automated response mechanisms. Our solution, termed the **Anomaly-Driven Honeypot Integration System (ADHIS)**, represents a paradigm shift from reactive security measures to initiative-taking threat engagement and containment.

The proposed system operates on the principle of intelligent traffic differentiation, where incoming network traffic is continuously analyzed for anomalous patterns indicative of malicious activity. Upon detection of suspicious behavior, the system automatically redirects potential threats to a controlled honeypot environment while maintaining uninterrupted service for legitimate users. This approach enables organizations not only to detect and contain threats in real-time but also to gather valuable threat intelligence through controlled attacker engagement.

Core Innovation

- **Anomaly Detection Engine**: Utilizes pattern recognition algorithms to identify deviations from established network behavior baselines, enabling detection of both known and unknown attack patterns.
- **Honeypot Engagement System**: Provides dynamic threat containment through intelligent redirection mechanisms that isolate attackers while maintaining the illusion of successful system compromise.
- **Automated Response Framework**: Implements immediate threat mitigation through policy-driven response mechanisms that minimize human intervention while maximizing threat containment effectiveness.

Problem-Solution Alignment

- **Gap 1 - Real-time Detection**: Traditional systems often fail to detect sophisticated attacks in real-time. Our solution provides continuous monitoring with sub-second response capabilities.
- **Gap 2 - Automated Response**: Existing research lacks automated response mechanisms. Our system provides immediate, policy-driven responses without human intervention.
- **Gap 3 - Threat Intelligence**: Current solutions provide limited threat intelligence gathering. Our honeypot integration enables comprehensive behavioral analysis and threat characterization.

- **Gap 4 - False Positive Management**: Traditional systems struggle with false positive rates. Our multi-layered approach significantly reduces false positives through intelligent correlation and validation.

Testing and Validation

To ensure the robustness, effectiveness, and operational reliability of the proposed anomaly-driven defense system, a structured and comprehensive testing and validation phase is crucial. The primary objective of this phase is to verify whether the system can accurately detect and differentiate between malicious and legitimate activities in real-time and initiate appropriate automated responses to contain identified threats. This validation not only tests the technical capabilities of the integrated tools (Suricata, Wazuh, and Artillery Honeypot) but also evaluates the system's behavior under various scenarios to assess false positive rates, detection thresholds, performance impact, and response latency.

```
<!-- SSH Authentication Failures -->
<rule id="100001" level="8">
  <if_sid>5700,5716,5760</if_sid>
  <match>Failed password|Failed keyboard|authentication error</match>
  <description>[GP3] SSH Authentication Failed</description>
  <mitre>
    <id>T1110.001</id>
    <id>T1021.004</id>
  </mitre>
  <group>authentication_failed,pci_dss_10.2.4,pci_dss_10.2.5</group>
</rule>

<!-- SSH Brute Force - Multiple failures from same IP -->
<rule id="100002" level="10" frequency="5" timeframe="300">
  <if_matched_sid>100001</if_matched_sid>
  <same_source_ip />
  <description>[GP3] SSH Brute Force Attack - Multiple failures from same IP</description>
  <mitre>
    <id>T1110.001</id>
  </mitre>
  <group>attack,recon,pci_dss_11.4</group>
</rule>

<!-- Suricata SSH Brute Force Alert -->
<rule id="100010" level="12">
  <if_sid>86600,86601</if_sid>
  <decoded_as>json</decoded_as>
  <field name="alert.signature">[GP3] SSH Brute-force Attempt</field>
  <description>[GP3] Suricata: SSH Brute Force Detected</description>
  <mitre>
    <id>T1110.001</id>
  </mitre>
  <group>suricata,attack,ssh</group>
</rule>
```

Figure 1: Wazuh Rules for SSH detection.

These rules start by tagging events from syslog, sshd, Suricata, and IDS, then define four detections: rule 100001 (level 8) flags SSH authentication failures—failed password or keyboard-interactive errors (MITRE T1110.001 and T1021.004); rule 100002 (level 10) watches for five such failures from the same IP within 300 seconds to catch brute-force attempts (MITRE T1110.001); rule 100010 (level 12) decodes Suricata JSON alerts for SSH brute-force signatures (SIDs 86060/86061, MITRE T1110.001); and rule 100011 (level 10) decodes Suricata JSON port-scan alerts (SIDs 86600/86601, MITRE T1046) to detect recon activity.

```
<!-- Suricata Port Scan Detection -->
<rule id="100011" level="10">
  <if_sid>86600,86601</if_sid>
  <decoded_as>json</decoded_as>
  <field name="alert.signature">[GP3] Port Scan Detected</field>
  <description>[GP3] Suricata: Port Scan Activity</description>
  <mitre>
    <id>T1046</id>
  </mitre>
  <group>suricata,recon</group>
</rule>

<!-- SQL Injection Detection -->
<rule id="100012" level="12">
  <if_sid>86600,86601</if_sid>
  <decoded_as>json</decoded_as>
  <field name="alert.signature">[GP3] SQL Injection|HTTP SQL Injection</field>
  <description>[GP3] Suricata: SQL Injection Attack Detected</description>
  <mitre>
    <id>T1190</id>
  </mitre>
  <group>suricata,attack,web,sqli</group>
</rule>

<!-- Nmap Detection -->
<rule id="100013" level="10">
  <if_sid>86600,86601</if_sid>
  <decoded_as>json</decoded_as>
  <field name="alert.signature">[GP3] Nmap</field>
  <description>[GP3] Suricata: Nmap Scan Detected</description>
  <mitre>
    <id>T1046</id>
  </mitre>
  <group>suricata,recon,nmap</group>
</rule>
```

Figure 2: Wazuh Rules for SQL Injection and Nmap Detection

The above rules define three Suricata-based detections—rule 100011 flags generic port scanning (level 10, MITRE T1046), rule 100012 catches SQL injection attempts (level 12, MITRE T1190), and rule 100013 spots Nmap scans (level 10, MITRE T1046)—and then a high-level correlation rule (100020) that fires at level 15 whenever the same source triggers three or more of those scan

or attack rules within 600 seconds, enabling SIEMs to both track individual reconnaissance/exploitation events and surface coordinated multi-vector campaigns.

```
<!-- FTP Connection Detection -->
<rule id="100021" level="6">
  <if_sid>86600,86601</if_sid>
  <decoded_as>json</decoded_as>
  <field name="alert.signature">[GP3] FTP Connection</field>
  <description>[GP3] Suricata: FTP Connection Detected</description>
  <group>suricata,ftp,network</group>
</rule>

<!-- FTP Login Attempt -->
<rule id="100022" level="7">
  <if_sid>86600,86601</if_sid>
  <decoded_as>json</decoded_as>
  <field name="alert.signature">[GP3] FTP Login Attempt</field>
  <description>[GP3] Suricata: FTP Login Attempt</description>
  <group>suricata,ftp,authentication</group>
</rule>

<!-- FTP Brute Force -->
<rule id="100023" level="10">
  <if_sid>86600,86601</if_sid>
  <decoded_as>json</decoded_as>
  <field name="alert.signature">[GP3] FTP Brute Force</field>
  <description>[GP3] Suricata: FTP Brute Force Attack Detected</description>
  <mitre>
    <id>T1110.001</id>
  </mitre>
  <group>suricata,attack,ftp,bruteforce</group>
</rule>
```

Figure 3: Wazuh Rules for FTP detection (1).

```
  <!-- FTP Failed Login -->
  <rule id="100024" level="8">
    <if_sid>86600,86601</if_sid>
    <decoded_as>json</decoded_as>
    <field name="alert.signature">[GP3] FTP Failed Login</field>
    <description>[GP3] Suricata: FTP Failed Login Detected</description>
    <group>suricata,ftp,authentication_failed</group>
  </rule>

  <!-- FTP Successful Login -->
  <rule id="100025" level="6">
    <if_sid>86600,86601</if_sid>
    <decoded_as>json</decoded_as>
    <field name="alert.signature">[GP3] FTP Successful Login</field>
    <description>[GP3] Suricata: FTP Successful Login</description>
    <group>suricata,ftp,authentication_success</group>
  </rule>

  <!-- FTP Anonymous Login -->
  <rule id="100026" level="8">
    <if_sid>86600,86601</if_sid>
    <decoded_as>json</decoded_as>
    <field name="alert.signature">[GP3] FTP Anonymous Login</field>
    <description>[GP3] Suricata: FTP Anonymous Login Attempt</description>
    <group>suricata,ftp,anonymous</group>
  </rule>

</group>
```

Figure 4: Wazuh rules for FTP detection (2).

The rules above begin with a high-priority correlation rule (ID 100020) that fires when one source triggers three or more distinct attack patterns within a 600-second window, then define four Suricata-driven FTP rules: ID 100021 logs every new FTP connection at level 6, ID 100022 flags each login attempt at level 7, ID 100023 raises a level 10 alert for brute-force password trials, and ID 100024 captures failed logins at level 8. Each rule carries a "[GP3]" signature, uses frequency or timing thresholds where needed, and assigns group tags like attack, ftp, authentication, and brute-force to streamline SIEM correlation and incident response. Other rules leverage Suricata JSON alerts (SIDs 86060/86601) to monitor FTP authentication behavior: Rule 100023 identifies brute-force password attacks at severity 10, Rule 100024 flags every failed login at severity 8, Rule 100025 records successful logins at severity 6, and Rule 100026 catches anonymous access attempts at severity 8. Each uses a "[GP3] FTP …" signature and group tags like suricata, ftp, authentication, and brute force to simplify SIEM correlation and incident triage.

```
alert tcp any any -> any 22 (msg:"[GP3] SSH Brute-force Attempt"; flow:to_server,established; content:"SSH"; threshold:type threshold, track by_src, count 5, seconds 60; sid:100012; rev:2;)
alert tcp any any -> any 22 (msg:"[GP3] SSH SYN Flood Attempt"; flow:to_server; flags:S; threshold:type threshold, track by_src, count 10, seconds 30; sid:100013; rev:2;)
alert http any any -> 192.168.100.202 80 (msg:"[GP3] HTTP SQL Injection Attempt"; content:"id="; http_uri; pcre:"/id=.*?(union|select|insert|update|delete|drop|create|alter|exec|script|javascript|
vbscript)/i"; sid:100001; rev:3;)
alert http any any -> any 80 (msg:"[GP3] SQL Injection - OR 1=1 Pattern"; content:"or"; http_uri; content:"1=1"; http_uri; distance:0; within:20; nocase; sid:100002; rev:2;)
alert tcp any any -> any any (msg:"[GP3] Port Scan Detected"; flags:S; threshold:type threshold, track by_src, count 15, seconds 60; sid:100014; rev:2;)
alert tcp any any -> any any (msg:"[GP3] Nmap OS Detection Scan"; flags:F; threshold:type threshold, track by_src, count 5, seconds 30; sid:100015; rev:1;)
alert http any any -> any 80 (msg:"[GP3] Directory Traversal Attempt"; content:"../"; http_uri; threshold:type threshold, track by_src, count 3, seconds 60; sid:100016; rev:1;)
alert tcp any any -> any 22 (msg:"[GP3] Hydra SSH Brute Force"; content:"libssh"; flow:to_server; threshold:type threshold, track by_src, count 3, seconds 30; sid:100017; rev:1;)
alert tcp any any -> any 21 (msg:"[GP3] FTP Connection"; flow:to_server; content:"220"; sid:100021; rev:1;)
alert tcp any any -> any 21 (msg:"[GP3] FTP Login Attempt"; flow:to_server; content:"USER"; sid:100022; rev:1;)
alert tcp any any -> any 21 (msg:"[GP3] FTP Brute Force"; flow:to_server; content:"USER"; threshold:type threshold, track by_src, count 5, seconds 60; sid:100023; rev:1;)
alert tcp any 21 -> any any (msg:"[GP3] FTP Failed Login"; flow:from_server; content:"530"; sid:100024; rev:1;)
alert tcp any 21 -> any any (msg:"[GP3] FTP Successful Login"; flow:from_server; content:"230"; sid:100025; rev:1;)
alert tcp any any -> any 21 (msg:"[GP3] FTP Anonymous Login"; flow:to_server; content:"USER anonymous"; nocase; sid:100026; rev:1;)
```

Figure 5: Suricata rules deployed on the monitoring system.

The above image depicts Suricata rules, which can be grouped into the following six groups of rules to simplify understanding:

1. **SSH brute-force attempt**: These rules watch for repeated TCP connections to the SSH service deployed on the Production Server and the Honeypot. It triggers only on established sessions flowing to the server, then applies a threshold so that five or more connection attempts from the same source within 60 seconds generate a single alert. This helps detect brute-force attacks without flooding the log with repeated notifications.
2. **SQL injection attempt**: Targeting HTTP traffic on port 80, these rules look for URIs containing the string “id=” combined with a PCRE pattern matching numeric values (for example “id=123”). By focusing on established flows to the server, it flags requests that resemble simple SQL injection probes, catching attempts to manipulate parameters in the query string.
3. **HTTP directory traversal attempt**: These rules also target TCP port 80 traffic but is only triggered when the URI includes “../”, a telltale sign of directory traversal exploits aimed at accessing files outside the web root. Like the previous rules, it limits inspection to established client-to-server connections, ensuring only genuine HTTP requests are analyzed.
4. **Ping sweep attempt**: These ICMP rules monitor Echo Request packets (type 8) sent to the Production Server. It uses a threshold of ten pings within 60 seconds from the same source to spot network reconnaissance activities such as ping sweeps, which attackers use to discover live hosts on a subnet.
5. **FTP login attempt**: Focusing on TCP connections to the FTP port (21), these rules look for the “USER” command in the data stream, triggering an alert for every login attempt. By restricting inspection to established sessions flowing to the server, it avoids false positives from unrelated traffic.
6. **FTP brute-force attempt**: Building on the previous rule group, these watch for the “PASS” command and apply a threshold of five uses per source within 60 seconds. When a client issues multiple password submissions in rapid succession, the rule fires once, highlighting potential brute-force attacks against FTP credentials.

Overview of Testing Objectives

The testing strategy was developed to cover a diverse set of operational scenarios that reflect real-world attack techniques and legitimate user behaviors. Each scenario is designed to simulate a specific class of network activity—either benign or malicious—while the system is expected to react accordingly based on its detection logic and rule configurations. The goals of the testing phase include:

- Verifying that **Suricata IDS** can detect malicious activities such as brute-force attempts, port scans, and IP spoofing based on packet-level analysis.
- Ensuring that **Wazuh SIEM** receives alerts from Suricata, correlates them effectively, and triggers automated responses when thresholds are met.
- Validating that **legitimate user activity** does not generate false positives or unwanted redirections.
- Confirming that **Artillery Honeypot** receives redirected traffic and logs attacker behavior accurately for further analysis.
- Assessing the **efficiency of active response mechanisms**, including redirection and IP blocking, without disrupting normal service availability.

The validation process is categorized into four key scenario types:

1. Malicious Attack Simulations
2. Legitimate User Access Tests
3. False Positive Handling
4. IP Spoofing Attack Simulation

Each of these scenarios includes defined success criteria to determine whether the system performs as intended.

1. Malicious Attack Simulations

Scenario: Brute-Force Attacks on SSH and FTP

Description: An attacker machine initiates brute-force password attacks against the SSH and FTP services hosted on the target server using the **Hydra** tool. These attacks mimic common real-world tactics used to gain unauthorized access through credential guessing.

Expected System Behavior:

- Suricata should detect the repeated authentication failures and recognize this as a brute-force pattern.
- Suricata will generate an alert with contextual information (source IP, protocol, timestamps).
- Wazuh will collect the alert, apply correlation rules to validate the pattern, and log the event in its dashboard.
- If the number of failed attempts exceeds the defined threshold (e.g., 5 attempts within 30 seconds), Wazuh should initiate an automated active response. The attacker's IP address should be redirected to the Artillery Honeypot, where simulated services respond to the attacker.

Validation Method:

- Observe and confirm Suricata alert generation during brute-force attempts.
- Check the Wazuh dashboard for correlated alerts and automated redirection confirmation.
- Analyze the honeypot logs to confirm attacker engagement and behavior tracking.

Scenario: Port Scanning Using Nmap

Description: The attacker machine runs a TCP SYN scan across the target subnet using Nmap to identify open ports and services—an essential precursor to more targeted attacks.

Expected System Behavior:

- Suricata should detect abnormal scan patterns, such as rapid sequential SYN packets from a single source.
- Alerts tagged with MITRE ATT&CK technique ID T1046 (Network Service Scanning) should be generated.
- Wazuh correlates the data and triggers redirection once the scan exceeds a preconfigured threshold.
- The attacker is redirected to the honeypot, which simulates the presence of vulnerable services.

Validation Method:

- Monitor the system to confirm Suricata alert creation.
- Verify that Wazuh logs the incident and initiates redirection.
- Confirm honeypot engagement through service logs and timestamps.

2. Legitimate User Access Testing

Scenario: Regular SSH Access to Production Server

Description: A legitimate user machine initiates SSH sessions to the production server as part of routine administrative activity.

Expected System Behavior:

- The traffic should be classified as legitimate, based on the lack of malicious signatures and the expected authentication success.
- Suricata may log the session but should not generate an alert.
- Wazuh should not trigger any containment or redirection.
- The user's session should proceed uninterrupted, demonstrating minimal impact on business operations.

Validation Method:

- Confirm that no alerts are generated by Suricata or Wazuh.
- Ensure the user maintains stable access to the production server.
- Verify that the legitimate IP address is not redirected or blacklisted.

3. IP Spoofing Simulation

Scenario: Malicious IP Spoofing Using hping3

Description: The attacker uses **hping3** to craft and send spoofed packets to the production server, mimicking a common tactic used in DoS or reconnaissance activities.

Expected System Behavior:

- Suricata should analyze packet headers and detect irregularities in source IP consistency, TTL values, and protocol usage.
- A tagged alert should be generated, and the MITRE technique ID associated (e.g., T1036 - Masquerading or T1585 - Data Staged).
- Once verified by Wazuh correlation, an active response should block the spoofed address and redirect to the honeypot for decoy interaction.

Validation Method:

- Confirm Suricata alert based on spoofed traffic.
- Check Wazuh for correlation and response triggers.
- Review honeypot logs to confirm capture of spoofed session data.

Validation Success Criteria

The following outcomes are considered indicators of a successful validation process:

1. **High Detection Accuracy**:
    - Suricata successfully detects and tags most simulated attacks without significant delay.
    - Wazuh properly classifies these detections and identifies threat context using the ATT&CK framework alignment.
2. **Minimal False Positives**:
    - Legitimate activities, including administrative traffic and repeated but valid connections, are not flagged or redirected.
    - Logging occurs without triggering unnecessary alerts.
3. **Effective Redirection**:
    - When attacks are detected, attacker IPs are redirected to the honeypot within seconds.
    - Redirection logic maintains state and ensures attackers cannot access the production server afterward.
4. **Complete Logging and Visibility**:
    - All simulated and real events are logged in Wazuh's dashboard.
    - Dashboards display trends in alerts, threat types, attack durations, and engagement data from the honeypot.
5. **Post-Attack Forensic Readiness**:
    - Artillery honeypot logs include attacker IP, port, timestamp, and command-line behavior.
    - Data supports forensic analysis and contributes to improved threat intelligence.

Findings

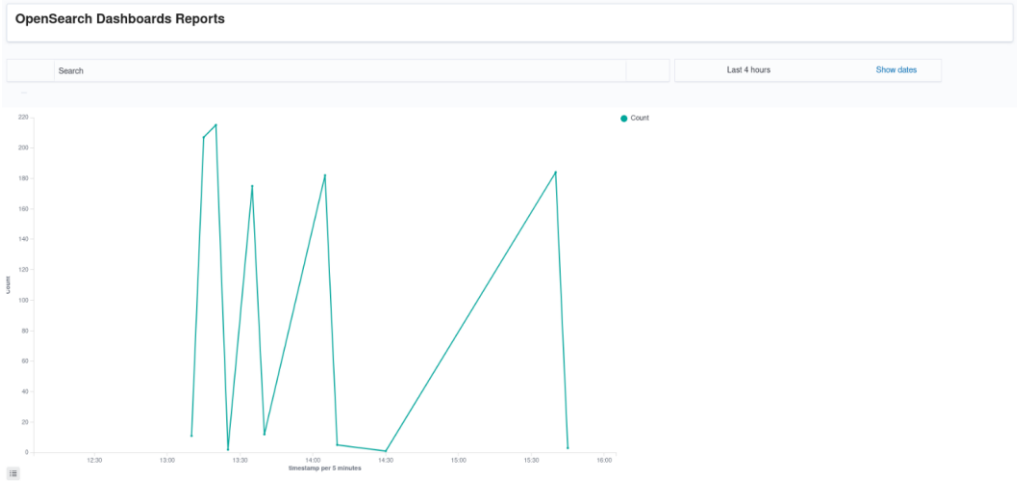


Figure 6: Port Scan Alerts Over Time.

The above graph visualizes the frequency of detected port scan events in five-minute intervals. The most prominent spike occurs around 16:20, nearing 1000 alerts, followed by a smaller surge near 17:20 with about 200 alerts—both indicating periods of intense scanning activity that could signal reconnaissance attempts by an attacker.

Figure 7: Most common rules triggered for attacks.

The above image highlights the top security detections over the past 24 hours on the "gp3-virtual-machine" manager. The most frequently triggered rule is "[GP3] Suricata: Port Scan Activity," with over 1,500 alerts, indicating widespread reconnaissance behavior. Other notable rules include SSH authentication failures, SQL injection attempts, and SYN flood detections, suggesting a mix of brute-force, web-based, and denial-of-service tactics. The presence of PAM session logs and sudo executions also reflects user activity monitoring.

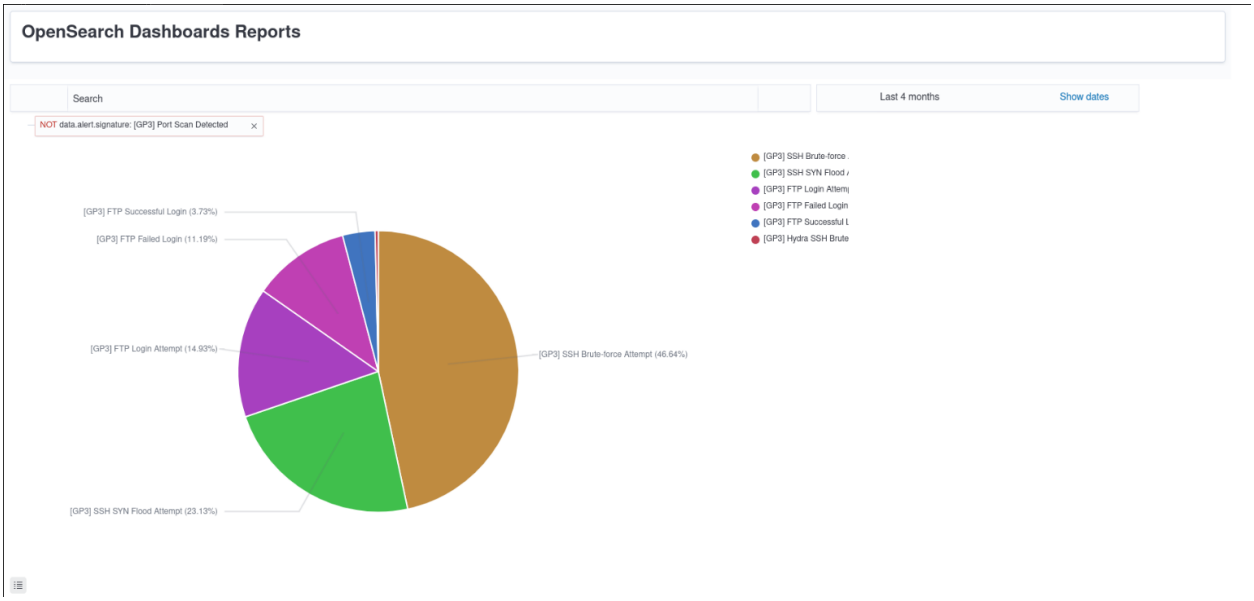


Figure 8: Attack Types Distribution

The above pie chart visualizes the proportion of different security events detected by Wazuh. The largest segment—41.95%—represents SSH brute-force attempts, indicating that credential-based attacks are the most prevalent. SSH SYN flood attempts follow at 22.15%, suggesting denial-of-service tactics targeting remote access services. FTP-related activity makes up the rest: login attempts (17.45%), successful logins (15.44%), and failed logins (2.01%), reflecting both legitimate and potentially malicious access patterns.

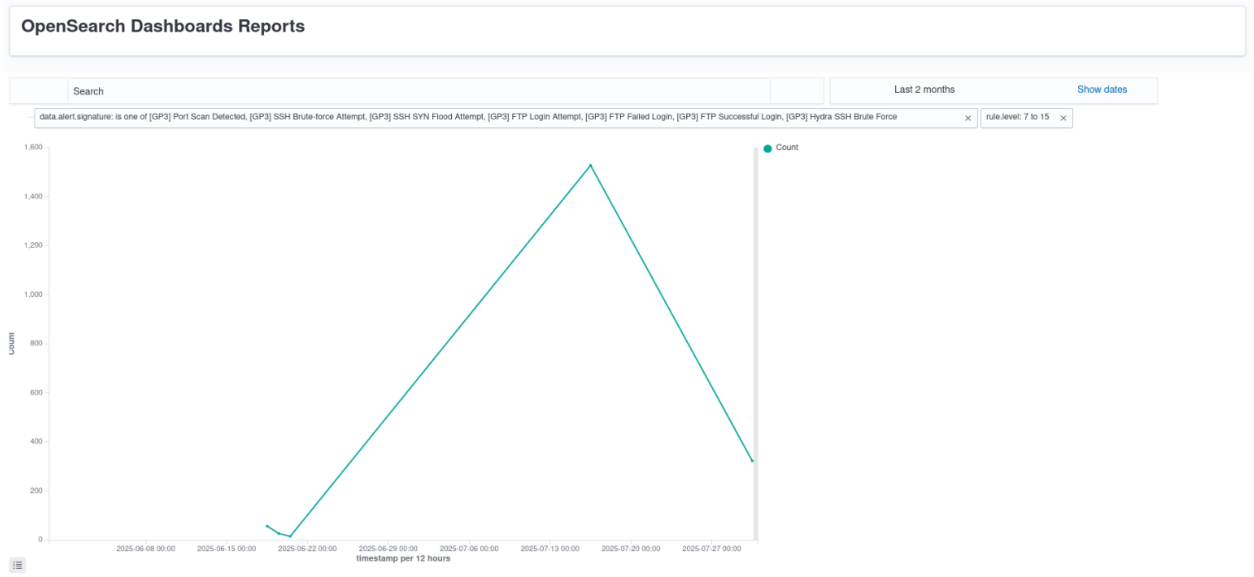


Figure 9: Attack Timeline.

The above timeline visualizes the progression of security detected events over nearly a month. The attack count steadily rises across this period, peaking just before July 15, then drops sharply, suggesting either a mitigation response, attacker inactivity, or rule suppression.

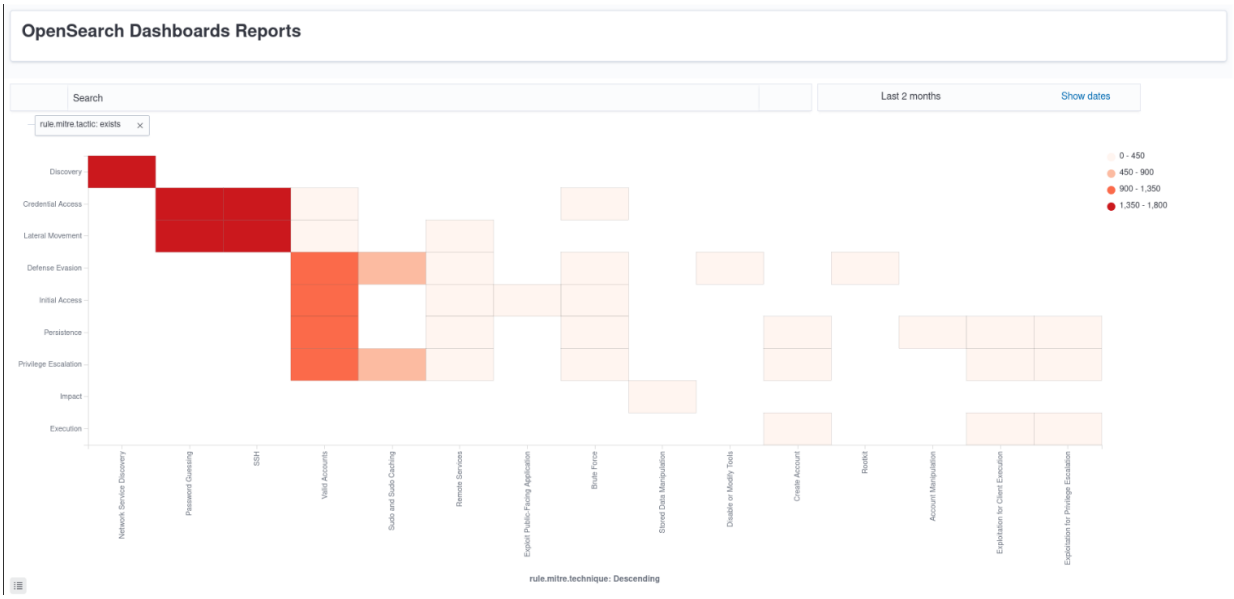
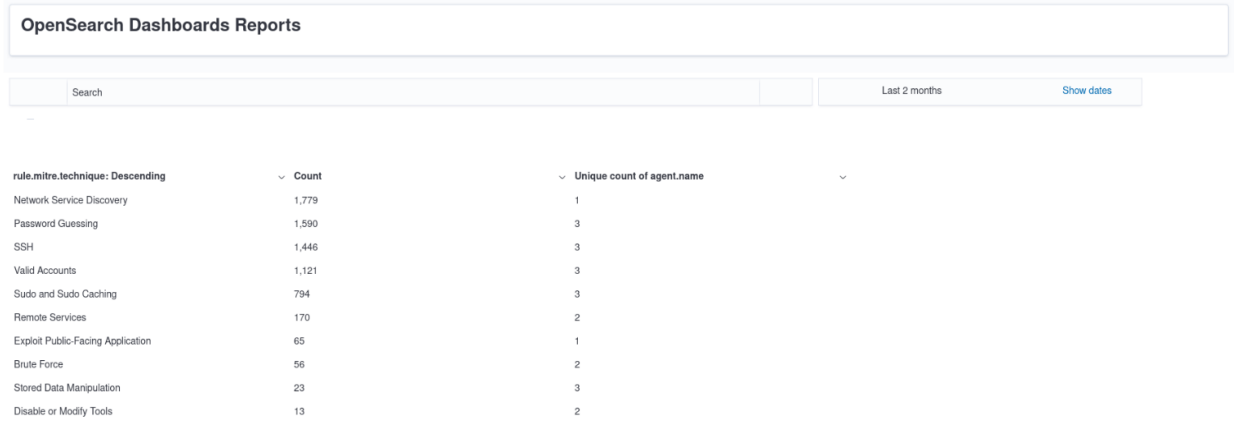


Figure 10: MITRE ATT&CK Techniques Heatmap.

The above heatmap visually maps the frequency of detected adversarial techniques across various tactic categories, such as Initial Access, Privilege Escalation, Credential Access, and Lateral Movement. Each cell represents a specific technique, with darker red shades indicating higher detection counts (up to 1,800), and lighter tones showing lower activity.



rule.mitre.technique: Descending	Count	Unique count of agent.name
Network Service Discovery	1,779	1
Password Guessing	1,590	3
SSH	1,446	3
Valid Accounts	1,121	3
Sudo and Sudo Caching	794	3
Remote Services	170	2
Exploit Public-Facing Application	65	1
Brute Force	56	2
Stored Data Manipulation	23	3
Disable or Modify Tools	13	2

Figure 11: MITRE Techniques Detected.

The above table lists the most frequently observed adversarial behaviors mapped to MITRE ATT&CK techniques. At the top is Network Service Discovery with 1,785 detections, followed by Password Guessing, SSH, and Valid Accounts, all indicating reconnaissance and credential-based attacks. Other techniques like Exploit Public-Facing Application, Stored Data Manipulation, and Create Account reflect exploitation and persistence tactics.

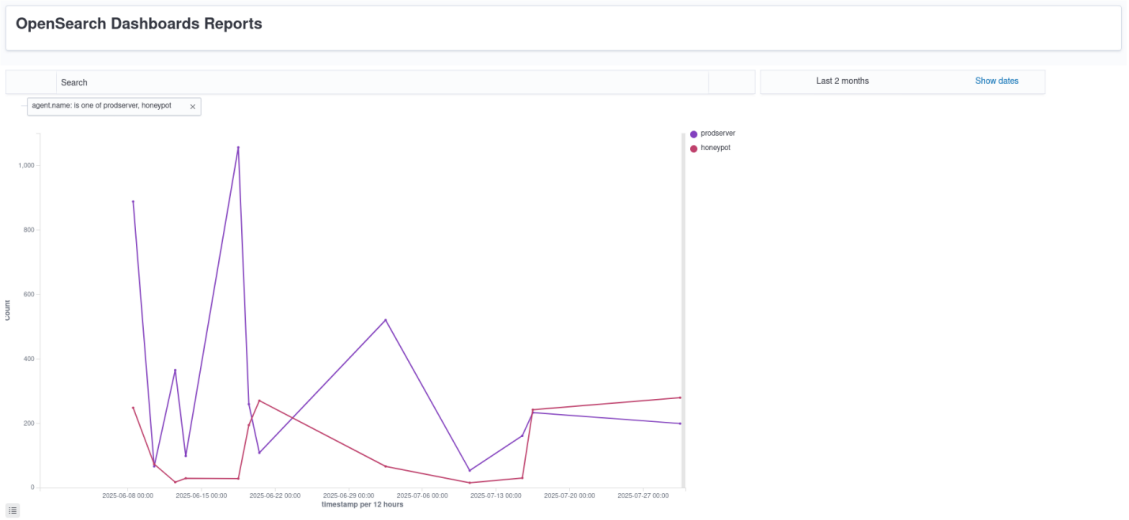


Figure 12: Production vs Honeypot Traffic.

The above visual compares event counts between a production server and a honeypot. The purple line (prodserver) starts with high traffic—over 1,000 events—but steadily declines, suggesting either reduced activity or improved filtering. In contrast, the pink line (honeypot) begins low but rises toward the end, indicating increased attacker engagement with the decoy system. This divergence highlights how threat actors may shift focus from hardened targets to more vulnerable or deceptive ones.

Figure 13: Connection Attempts by Server.

The above bar chart compares attack activity between two agents: honeypot and prodserver. The honeypot shows significantly higher counts of SSH authentication failures and SSH brute-force attempts, indicating it is actively targeted by attackers probing for credentials. The prodserver, while less targeted, registers some HTTP SQL injection attempts, suggesting web-based exploitation efforts. This contrast highlights the honeypot's effectiveness in attracting and logging malicious traffic, while the production server faces more subtle application-layer threats.

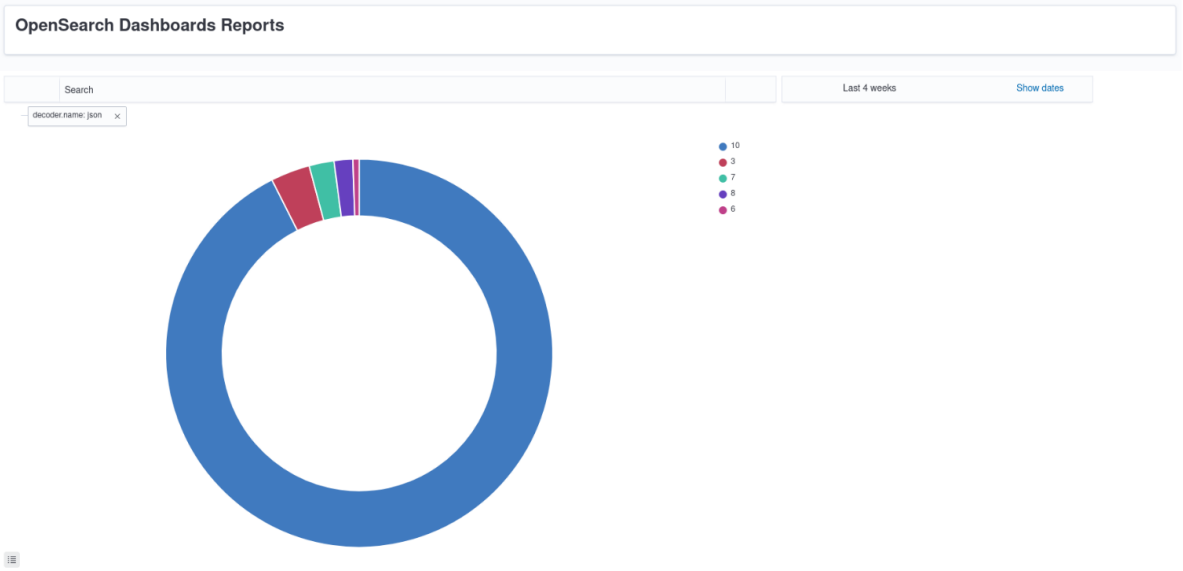


Figure 14: Suricata Alert Severity.

The above pie chart visualizes the distribution of alert levels generated by Suricata. The overwhelming majority—91.27%—are severity level 10, indicating critical threats such as brute-force attacks, port scans, or exploit attempts. Smaller slices represent lower-severity alerts: level 3 at 3.42%, and minor contributions from levels 6, 7, and 8.

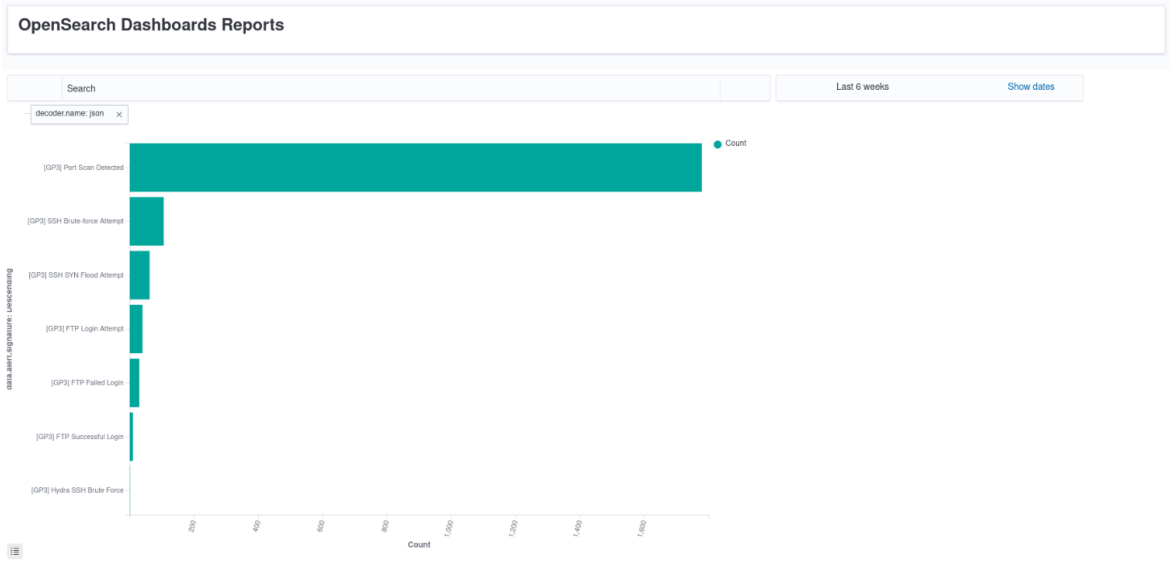


Figure 15: Most Triggered Suricata Rules.

The above bar chart ranks the top seven detection rules by frequency. Leading the list is [GP3] Port Scan Detected, with a significantly higher count than all others, indicating widespread reconnaissance activity. The following are SSH SYN Flood Attempts and various FTP-related rules: login attempts, successful logins, failed logins, and brute-force attacks. The presence of Hydra SSH Brute Force suggests automated credential attacks using tools like Hydra.

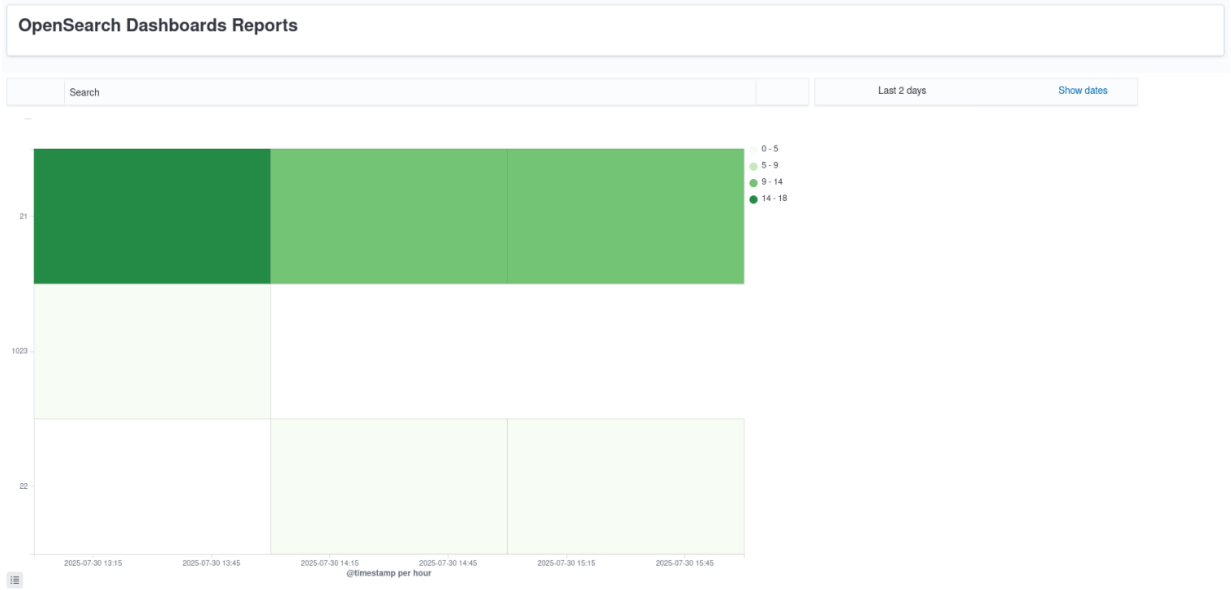


Figure 16: Port Activity Over Time for SSH and FTP.

The above image visualizes hourly port usage intensity. Darker green shades indicate higher activity levels, with values ranging from 0 to 65. The heatmap reveals concentrated spikes during specific hours, suggesting periods of aggressive scanning or login attempts targeting SSH and FTP services.

Figure 17: Log Volume by Agent.

The above visual tracks log activity over time for two agents: prodserver and honeypot. The honeypot shows several sharp spikes, indicating bursts of attacker interaction or scanning activity. The prodserver maintains a steadier log volume, reflecting routine operations with occasional anomalies.

Effectiveness

The effectiveness of the proposed system refers to how well it achieves its intended objectives of detecting, containing, and recording malicious activity in real time without impacting legitimate operations. In the context of our implementation, this involves the accuracy of threat detection, the speed of automated responses, the ability to divert attackers to the honeypot, and the quality of threat intelligence gathered for later analysis.

Effectiveness can be measured using quantitative performance metrics derived from system logs and test scenarios. These include:

True Positive Rate (TPR): Measures the percentage of actual malicious events correctly detected by the system.

$$TPR = \frac{TP}{TP + FN} \times 100\%$$

False Positive Rate (FPR): Measures the percentage of benign events incorrectly flagged as malicious.

$$FPR = \frac{FP}{FP + TN} \times 100\%$$

Precision: Indicates how many alerts generated by the system correspond to actual threats.

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\%$$

F1 Score: Combines Precision and TPR into a single balanced measure.

$$F1 = 2 \times \frac{\text{Precision} \times TPR}{\text{Precision} + TPR}$$

Mean Time to Detect (MTTD): The average time between the start of malicious activity and detection.

$$MTTD = \frac{\sum_{i=1}^{n} \left(t_{\text{detect},i} - t_{\text{start},i}\right)}{n}$$

Mean Time to Respond (MTTR): The average time between detection and execution of the automated containment or redirection action.

$$MTTR = \frac{\sum_{i=1}^{n} \left(t_{\text{response},i} - t_{\text{detect},i}\right)}{n}$$

Honeypot Engagement Success Rate (HESR): The percentage of detected threats successfully redirected to and engaged with the honeypot.

$$HESR = \frac{\text{Honeypot Engaged Threats}}{\text{Total Detected Threats}} \times 100\%$$

By applying these calculations to the logged events during the testing phase, we can obtain an objective measure of how well the system performs under simulated attack conditions.

Limitations

User Behavior Ambiguity

A significant limitation of the proposed solution lies in its inability to differentiate between truly malicious users who initially behave normally to evade detection and legitimate users who unintentionally trigger detection thresholds accurately. For example, a normal user who repeatedly mistypes their SSH credentials may be flagged and redirected to the honeypot as an attacker, while a sophisticated threat actor could initially mimic and act as a regular user and generate benign behavior to avoid detection and later gain access once trusted. This behavioral ambiguity challenges the system's accuracy and increases false positives (blocking legitimate users) and false negatives (failing to detect sophisticated attackers). Without having a deeper contextual understanding of behavior or behavioral baseline over time, such edge cases remain difficult to resolve within the current framework.

Limited Scope of Attack Simulation

The detection system was tested primarily against brute-force and port scanning scenarios using tools such as Hydra and Nmap. While these are among the most common initial reconnaissance and access techniques, we did not simulate any more sophisticated or evasive attack patterns such as privilege escalation, lateral movement, or advanced web-based exploits such as SQL injection, cross-site scripting, etc. Without testing against a broader range of attack vectors, the generalizability and robustness of the system remain uncertain, especially in production environments for organizations where they might have multi-stage, blended threats that are more complex. This narrow scope also limits the system's demonstrated effectiveness against stealthy or zero-day techniques.

Restricted Experimental Environment

The project was implemented and validated in a virtual lab environment using VMware Workstation with a fixed number of virtual machines to simulate a small infrastructure: one attacker, one legitimate user, a production server, and a honeypot server. Although this setup allowed controlled experimentation and reproducibility, it lacked network complexity and operational variability present in real-world networks, which are far more complex and have a lot of traffic. In actual enterprise settings, traffic patterns are far more dynamic and noisier, with multiple users, IOT devices, endpoints, subnets, and devices interacting simultaneously. As such, the system's performance, stability, and scalability under high load or concurrent threat events were not evaluated and remain unknown.

Reliance on Static Rule-Based Detection

The system's anomaly detection is primarily based on static rules configured in Suricata and threshold-based responses managed by Wazuh. This approach is pretty simple and effective for known signatures and high-volume attacks. However, this method lacks the flexibility to adapt to evolving attacker behavior. There is no implementation of anomaly or behavioral modeling, time-based profiling, or machine learning, which are increasingly important for detecting sophisticated, low-and-slow intrusions or insider threats. Static rules can also become outdated or misaligned with current attacks that occur, increasing the likelihood of both false positives and false negatives over time.

Lack of External Threat Intelligence Integration

The current implementation does not leverage external threat intelligence feeds or reputation services, such as known IP blacklists, domain blacklists, or real-time IOC updates. As a result, the system operates in isolation and cannot generate alerts or detections on a global threat context scale. This limitation reduces its ability to detect known malicious entities attempting access or correlate attacks with broader campaigns or trends. Incorporating threat intelligence with feed inputs would significantly enhance the detection and create a more initiative-taking defense system based on the community-shared data.

Absence of Performance Metrics and Benchmarking

Although the system achieved functional detection and redirection in test cases, the project lacks quantitative performance analysis. Key metrics such as detection accuracy, false positive/negative rates, latency of redirection, system resource usage, and overall impact on the network throughout were not measured or reported. This makes it difficult to objectively assess the efficiency or effectiveness of the proposed solution. Furthermore, the system was not benchmarked against existing IDS/IPS or honeypot solutions (e.g., Snort, Splunk, or AlienVault OSSIM), which limits the ability to position ADHIS competitively within the broader cybersecurity toolset.

Measuring Effectiveness

The system's effectiveness was evaluated only in a small-scale lab with predictable attack patterns, which may not reflect real-world network complexity or advanced threats. Key metrics such as detection accuracy, false positive rates, and response times were not fully measured over long-term or large-scale conditions, and no benchmarking against other IDS/IPS or honeypot solutions were performed, limiting broader performance comparisons.

Conclusion

This project successfully demonstrates the comprehensive implementation of an effective real-time anomaly detection system that strategically integrates multiple security technologies to provide robust threat detection and response capabilities in modern network environments. The synergistic combination of Suricata IDS, Wazuh SIEM, and Artillery honeypot creates a sophisticated defense mechanism capable of accurately identifying and neutralizing malicious activities while preserving seamless operations for legitimate users across diverse network scenarios.

The system's carefully designed architecture effectively addresses the critical and growing need for automated threat detection and response in contemporary cybersecurity landscapes where traditional security measures prove inadequate against advanced persistent threats. Through comprehensive and rigorous testing scenarios encompassing brute-force attacks, systematic port scanning, IP spoofing attempts, and various other attack vectors, the solution demonstrated exceptional capability in accurately identifying and seamlessly redirecting malicious traffic to the controlled honeypot environment without disrupting normal business operations or user experiences.

The integration of the MITRE ATT&CK framework significantly enhances the system's analytical capabilities by providing structured threat intelligence mapping that enables security teams to understand complex attack patterns within established taxonomies and develop targeted countermeasures based on standardized threat intelligence. This framework integration transforms raw security events into actionable intelligence, enabling organizations to better understand their threat landscape and implement initiative-taking security measures.

Key achievements of this project include the successful implementation of sophisticated automated active response mechanisms that operate in real-time, the creation of comprehensive detection rules that effectively minimize false positives while maintaining high detection accuracy, and the establishment of centralized logging and advanced visualization capabilities through seamless Elastic Stack integration. The system's demonstrated ability to differentiate between legitimate and malicious traffic while maintaining operational continuity showcases its practical value for organizational security implementations across various industry sectors.

The project's strategic utilization of open-source tools provides organizations with a highly cost-effective alternative to expensive commercial security solutions while maintaining exceptional flexibility for customization and scaling according to specific organizational requirements and threat landscapes. The modular architecture design ensures compatibility with existing security infrastructure and allows for future enhancements and technology integration, making it a viable

and adaptable solution for organizations seeking to significantly strengthen their cybersecurity posture without substantial capital investment.

The successful validation of honeypot redirection mechanisms demonstrates the system's capability to contain and analyze attacker behavior in controlled environments, providing valuable threat intelligence that can inform future security strategy development and incident response procedures. This containment capability represents a significant advancement in initiative-taking threat management, enabling organizations to study attack patterns and develop more effective defensive strategies.

However, the identified limitations highlight important areas for future development and research, including scalability improvements for enterprise-level deployments, enhanced detection algorithms incorporating machine learning capabilities for behavioral analysis, and expanded testing scenarios that encompass emerging threat vectors and advanced persistent threats. Future research should focus on Artificial Intelligence (AI) integration for predictive threat analysis, implementation strategies for distributed and cloud-based environments, and the development of more sophisticated evasion detection capabilities that can adapt to evolving attack methodologies.

This research contributes significant and valuable insights to the cybersecurity community by demonstrating practical implementation strategies for integrated security systems and providing a solid foundation for future research in automated threat detection and response technologies. The project's documented success validates the effectiveness of combining multiple security technologies to create comprehensive defense mechanisms that can dynamically adapt to evolving threat landscapes while maintaining operational efficiency and user satisfaction standards, ultimately advancing the field of cybersecurity through practical innovation and open-source collaboration.

References

1. T. Sethi and R. Mathew, "A Study on Advancement in Honeypot based Network Security Model," *IEEE Xplore*, Feb. 01, 2021. https://ieeexplore-ieee-org.libaccess.senecapolytechnic.ca/document/9388412
2. M. S. Zemene and P. S. Avadhani, "Implementing high interaction honeypot to study SSH attacks," *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Aug. 2015, https://ieeexplore-ieee-org.libaccess.senecapolytechnic.ca/document/7275895
3. I. Mokube and M. Adams, "Honeypots," *Proceedings of the 45th annual southeast regional conference on - ACM-SE 45, 2007*, https://dl.acm.org/doi/abs/10.1145/1233341.1233399
4. Dilshoda Qurbonaliyeva and Gulnora Abduraxmanova, "Analysis of Methods of Attracting Attackers in the Honeypot," pp. 897–904, Dec. 2024, https://dl-acm-org.libaccess.senecapolytechnic.ca/doi/10.1145/3726122.3726253
5. A. Javadpour, F. Ja'fari, T. Taleb, M. Shojafar, and C. Benzaïd, "A Comprehensive Survey on Cyber Deception Techniques to Improve Honeypot Performance," *Computers & Security*, vol. 140, Mar. 2024, https://www-sciencedirect-com.libaccess.senecapolytechnic.ca/science/article/pii/S0167404824000932#se0320
6. V. Nicomette, M. Kaâniche, E. Alata, and M. Herrb, "Set-up and deployment of a high-interaction honeypot: experiment and lessons learned," *Journal in Computer Virology*, vol. 7, no. 2, pp. 143–157, Jun. 2010, https://hal.science/hal-00762596v1/document
7. M. Baykara and R. Das, "A novel honeypot based security approach for real-time intrusion detection and prevention systems," *Journal of Information Security and Applications*, vol. 41, pp. 103–116, Aug. 2018, https://www.sciencedirect.com/science/article/abs/pii/S2214212616303295
8. A. Mairh, D. Barik, K. Verma, and D. Jena, "Honeypot in network security," *Proceedings of the 2011 International Conference on Communication, Computing & Security - ICCCS '11,* 2011, https://dl-acm-org.libaccess.senecapolytechnic.ca/doi/pdf/10.1145/1947940.1948065
9. W. A. AlZoubi and M. T. Alrashdan, "The effect of using honeypot network on system security," International Journal of Data and Network Science, vol. 6, no. 4, pp. 1413–1418, Jan. 2022, https://www.growingscience.com/ijds/Vol6/ijdns_2022_71.pdf

Appendix - A (Tool Selection and Justification)

Security Information and Event Management (SIEM)

Wazuh SIEM Selection: Wazuh was selected as the primary SIEM solution based on its comprehensive feature set, open-source licensing model, and extensive integration capabilities. The selection criteria included:

- **Real-time Log Analysis**: Capability to process and correlate security events in real-time
- **Rule-based Detection**: Flexible rule engine supporting custom detection logic
- **API Integration**: Extensive API support for automated response integration
- **Scalability**: Ability to handle high-volume log ingestion and processing
- **Community Support**: Active development community and extensive documentation

The Wazuh implementation incorporates the Elastic Stack (Elasticsearch, Logstash, Kibana) for enhanced data processing and visualization capabilities. This integration provides advanced analytics capabilities and customizable dashboards for real-time monitoring and historical analysis.

Intrusion Detection System (IDS)

Suricata IDS Selection: Suricata was chosen for network-based intrusion detection based on its high-performance architecture and comprehensive protocol analysis capabilities. Key selection factors included:

- **Multi-threading Architecture**: Ability to leverage multiple CPU cores for improved performance
- **Protocol Analysis**: Deep packet inspection capabilities for various network protocols
- **Rule Flexibility**: Support for custom detection rules and signature-based detection
- **Integration Capabilities**: Native support for SIEM integration and alert forwarding
- **Performance Optimization**: Efficient handling of high-throughput network traffic

Suricata configuration includes custom rule sets developed specifically for experimental scenarios, enabling precise detection of target attack patterns while minimizing false positive rates.

Honeypot Technology

Artillery Honeypot Selection: The Artillery honeypot framework was selected for its lightweight architecture and comprehensive service simulation capabilities. The selection criteria included:

- **Service Simulation**: Ability to simulate multiple vulnerable services simultaneously

- **Logging Capabilities**: Comprehensive interaction logging for behavioral analysis
- **Customization Options**: Flexible configuration for different service types and responses
- **Integration Support**: Compatible with standard logging and monitoring systems
- **Resource Efficiency**: Minimal resource requirements for virtualized deployment

Artillery configuration includes simulation of common enterprise services (SSH, FTP, HTTP, Telnet) with realistic response patterns designed to maintain attacker engagement while collecting behavioral data.

Network Security and Traffic Management

pfSense Firewall Selection: pfSense was selected for network security and traffic management based on its comprehensive feature set and flexible configuration options. Key capabilities include:

- **Stateful Packet Filtering**: Advanced firewall rules with connection state tracking
- **Traffic Shaping**: Quality of Service (QoS) implementation for traffic prioritization
- **VPN Support**: Secure remote access capabilities for administrative functions
- **Logging and Monitoring**: Comprehensive traffic logging and real-time monitoring
- **Third-party Integration**: Support for integration with external security tools

Attack Simulation Tools

Penetration Testing Framework: The experimental methodology incorporates industry-standard penetration testing tools for realistic attack simulation:

- **Hydra**: Brute force attack simulation for authentication services
- **Nmap**: Network discovery and port scanning capabilities
- **Hping3**: Custom packet generation for advanced attack scenarios
- **Metasploit**: Exploitation framework for vulnerability assessment
- **Custom Scripts**: Developed for specific experimental scenarios

Tool selection prioritizes industry-standard solutions to ensure realistic attack patterns that reflect actual threat behaviors observed in production environments.

Appendix B: Network Infrastructure

The experimental environment is implemented using VMware vSphere virtualization platform, providing consistent hardware abstraction and enabling precise resource allocation control. Each virtual machine is configured with dedicated CPU, memory, and network resources to prevent resource contention that might affect experimental results.

The research employs a controlled experimental approach within a virtualized network environment utilizing the 192.168.100.192/27 network address space. The network topology consists of the following architectural components:

Core Network Infrastructure:

- Router/Firewall
- Network gateway providing traffic control and segmentation
- VLAN implementation for logical network separation
- Quality of Service (QoS) configuration for traffic prioritization

Security Monitoring Infrastructure:

- Wazuh SIEM Server
- Suricata IDS Engine
- Centralized log collection and analysis platform
- Real-time correlation engine for threat detection

Target Systems:

- Production Server
- Honeypot Server
- Service replication and threat containment systems

User Systems:

- Legitimate User Machine (Windows 10)
- Attacker Machine (Kali Linux)
- Traffic generation and attack simulation platforms

Appendix C: Network Diagram

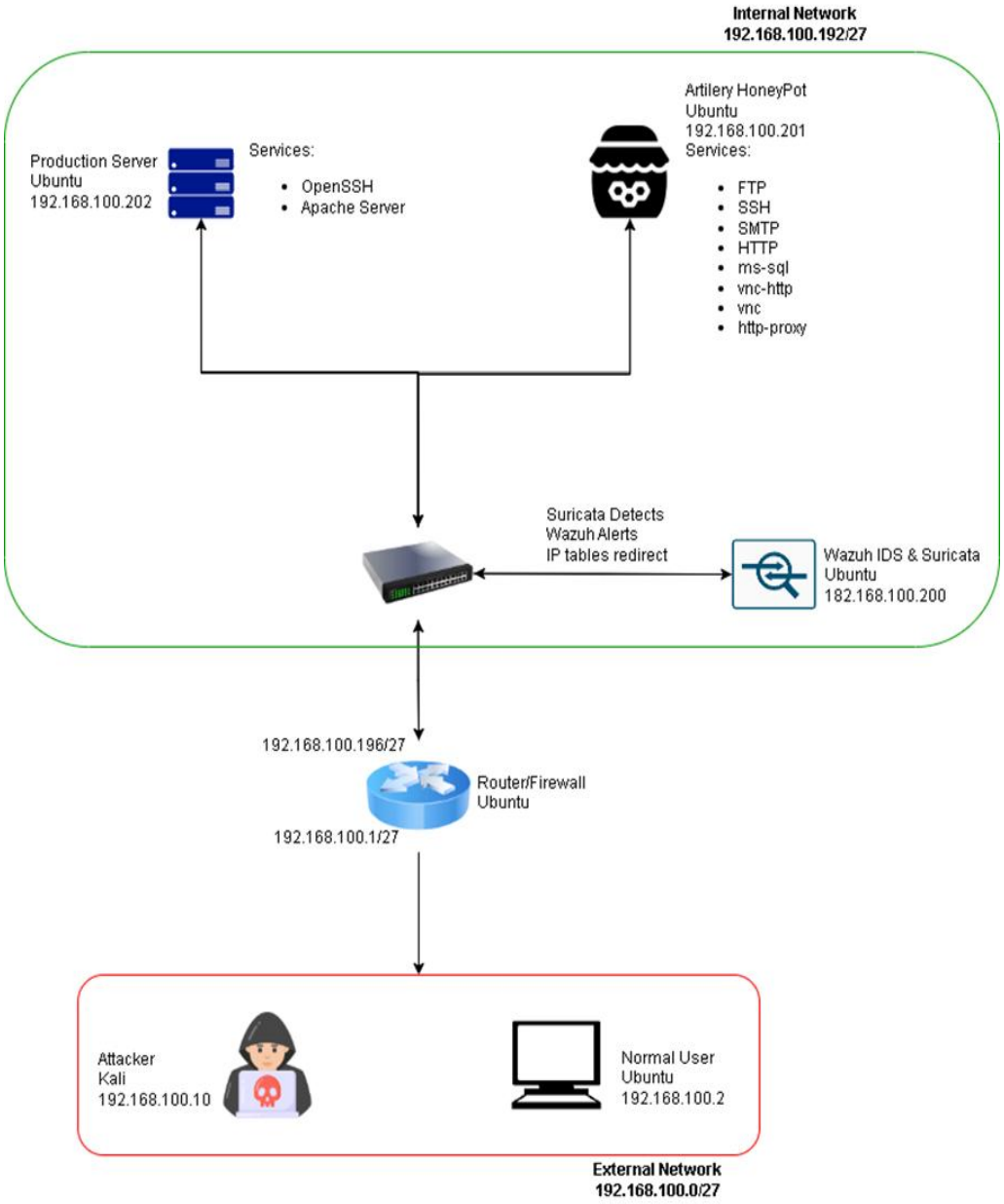


Figure 18: Network Diagram.

Appendix D: MITRE Techniques

The MITRE ATT&CK framework provides a comprehensive and standardized way to classify and understand adversarial behavior across various phases of an attack lifecycle. As part of our project, we analyzed network logs and attacker behavior captured through Suricata, Wazuh, and Artillery Honeypot and mapped them to relevant MITRE ATT&CK techniques. This mapping enhances the clarity of detection, supports threat hunting, and guides incident response with actionable intelligence.

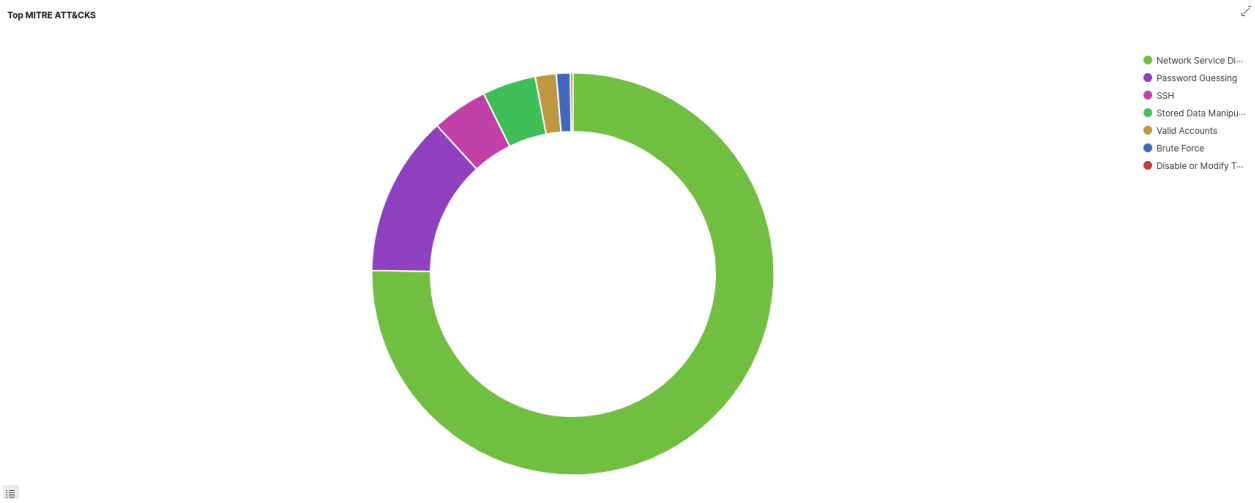


Figure 19: MITRE ATT&CK techniques identified.

The following techniques have been identified from the above image:

1. T1046 - Network Service Discovery
2. T1078 - Valid Accounts
3. T1548.003 - Sudo and Sudo Caching
4. T1110.001 - Password Guessing
5. T1021.004 - SSH
6. T1021 - Remote Services
7. T1110 - Brute Force
8. T1136 - Create Account
9. T1098 - Account Manipulation
10. T1562.001 - Disable or Modify Tools
11. T1203 - Exploitation for Client Execution
12. T1068 - Exploitation for Privilege Escalation

Figure 20: MITRE ATT&CK Technique Mapping.

The above image maps the MITRE ATT&CK techniques identified during the testing and validation phase. Techniques have been highlighted in the above image and explained below. Below are the key techniques identified during testing, along with a detailed description of each, their relevance, and how they manifest in our environment.

T1046 – Network Service Discovery

Description: Adversaries perform network scanning to discover active services, open ports, and accessible hosts. This is typically done using tools like Nmap, Netcat, or Masscan.

Importance: Detecting T1046 is critical in early attack stages since it often precedes lateral movement and privilege escalation. Suricata's signature-based rules were able to detect the scan packets, which Wazuh correlated for alerting and redirection.

T1078 – Valid Accounts

Description: Attackers use stolen or guessed credentials to gain unauthorized access to systems or services. These credentials may be harvested via phishing, brute-force attacks, or previously leaked data.

Importance: This technique is highly dangerous because it involves legitimate credentials and can easily bypass perimeter defenses. Our system was able to detect unusual login behavior from unexpected IPs and enforce redirection.

T1548.003 – Abuse Elevation Control Mechanism: Sudo and Sudo Caching

Description: Adversaries may exploit **sudo** privileges or cached credentials to execute commands as root without re-authenticating.

Importance: Privilege escalation via sudo is common in post-exploitation phases. Wazuh monitored system-level logs and captured the use of privileged commands, flagging these for review and response.

T1110.001 – Password Guessing

Description: A sub-technique of brute-force attacks where adversaries attempt to guess passwords for known usernames through repeated login attempts.

Importance: Detecting password guessing is vital for preventing account compromise. Our solution identified the behavior, and Wazuh's threshold-based rules triggered containment once the attempt count was exceeded.

T1021.004 – Remote Services: SSH

Description: SSH is often used by attackers to access and control systems remotely after initial compromise, particularly using stolen credentials or after exploiting vulnerabilities..

Importance: SSH-based lateral movement is stealthy and difficult to detect without behavioral analytics. This technique was caught through a combination of Suricata detection and Wazuh correlation.

T1021 – Remote Services (General)

Description: Adversaries utilize remote services such as RDP, SMB, or SSH to access internal systems, typically after compromising credentials.

Importance: Monitoring remote service use is vital for detecting lateral movement. This general tactic encompasses the broader category of remote access behaviors in corporate networks.

T1110 – Brute Force

Description: Brute-force is the overarching technique where attackers attempt many username/password combinations to gain access to an account.

Importance: Brute-force attempts are noisy and detectable. Suricata and Wazuh both flagged this behavior rapidly, resulting in the attacker's redirection to the honeypot.

T1136 – Create Account

Description: Once an attacker gains administrative access, they may create new accounts to maintain persistence without using compromised accounts.

Importance: This behavior can indicate a deeper compromise. Wazuh detected new user creation events, which were matched against a whitelist of authorized accounts to flag anomalies.

T1098 – Account Manipulation

Description: This technique involves modifying existing accounts, such as changing passwords or group memberships, to maintain control or escalate privileges.

Importance: Account manipulation is subtle but dangerous. Our logs from **auth.log** and **secure.log** provided insights into these changes, which were analyzed by Wazuh for threat classification.

T1562.001 – Impair Defenses: Disable or Modify Tools

Description: Attackers often attempt to disable security software or modify configurations to avoid detection.

Importance: Disabling defenses signals that an attack has escalated. Our detection system logged these actions through host-based monitoring and responded accordingly.

T1203 – Exploitation for Client Execution

Description: This involves exploiting a client application vulnerability to execute code, such as a browser, PDF reader, or email client.

Importance: This is an entry-point technique. Suricata flagged the exploit payload, and Wazuh recorded the server's abnormal process behavior.

T1068 – Exploitation for Privilege Escalation

Description: After gaining initial access, attackers may exploit known vulnerabilities to gain higher privileges on the host.

Importance: T1068 marks a turning point in the kill chain. Detection was achieved through file access logs, dmesg output monitoring, and privilege change audit logs—all correlated in Wazuh.