Layered Detection Engineering and Incident Analysis Using Wazuh SIEM

Author: Wayne Howlett
Date: May 05, 2026
Skill Area: Security Operations / Detection Engineering
Tools: Wazuh, Kali Linux, Hydra, Linux Server
Attack Type: SSH Brute Force (MITRE T1110)
Environment: VirtualBox Lab (Internal Network)
Outcome: Successfully implemented a layered detection strategy using Wazuh SIEM, introducing early warning and high-severity alerts through custom correlation rules to detect and escalate SSH brute-force activity.

1. Objective

The objective of this project was to extend an existing SIEM lab by designing and implementing layered detection logic for SSH brute-force attacks. Rather than relying solely on default Wazuh rules, this project focuses on building custom correlation-based detection that identifies suspicious behavior early and escalates alerts as attack activity progresses.

In addition to generating alerts, this project aimed to deepen understanding of how detection rules operate within a SIEM, how multiple events can be correlated to identify meaningful patterns, and how threshold-based logic can be tuned to better reflect real-world attack scenarios. The goal was to move beyond basic log monitoring and develop a more analytical, behavior-based detection approach.

---

2. Scenario

Brute-force attacks are a widely used technique in which attackers attempt to gain unauthorized access by repeatedly trying different password combinations against a target system. These attacks are often automated and can generate a high volume of authentication attempts in a very short period. Services such as SSH are common targets due to their remote access capabilities.

In many cases, individual failed login attempts may appear harmless or may be attributed to user error. However, when analyzed collectively, repeated failures from a single source can indicate malicious intent. This creates a challenge for detection systems, which must distinguish between normal activity and coordinated attack behavior.

In this scenario, a controlled brute-force attack was simulated using a Kali Linux system targeting a Linux server. Wazuh SIEM was configured to monitor authentication logs and analyze login behavior. The goal was to replicate a real-world situation in which a security analyst must detect and interpret patterns across multiple events rather than relying on isolated alerts.

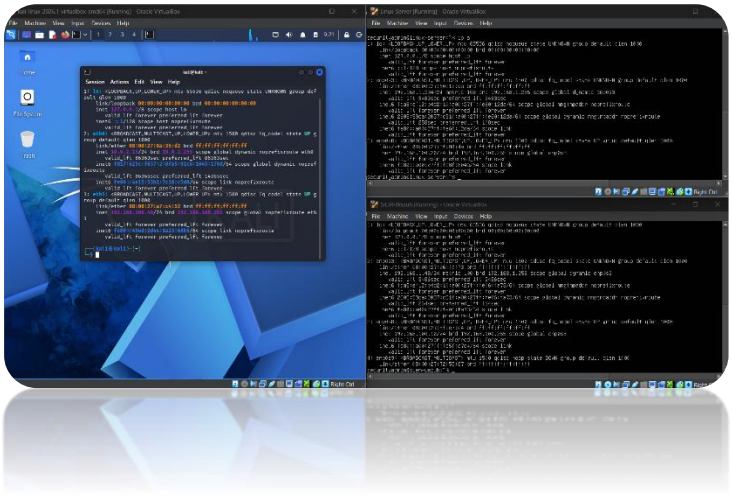
---

3. Environment

The lab environment was built using VirtualBox and consisted of multiple virtual machines connected through an isolated internal network. The Wazuh SIEM server was deployed on a dedicated system and configured to collect logs from the Linux target machine.

The Linux server acted as the primary target, specifically through SSH authentication. A Kali Linux system was used as an attacker to simulate brute-force login attempts. All systems were configured within the same subnet to ensure reliable communication and controlled testing conditions.

Figure B1. Lab environment and network configuration



4. Tools Used

This project utilized several tools to simulate attack activity and perform detection analysis. Wazuh SIEM served as the central monitoring platform, responsible for collecting logs, applying detection rules, and generating alerts based on defined conditions. Its ability to process and correlate events made it ideal for implementing layered detection logic.

Kali Linux was used as the attacker platform, providing access to a wide range of penetration testing tools. Hydra, a password-cracking utility included in Kali, was specifically used to simulate brute-force attacks against the SSH service. The Linux server acted as the target system, generating authentication logs that were forwarded to Wazuh for analysis.

Together, these tools created a realistic testing environment that allowed for both attack simulation and detection engineering within a controlled lab setup.

5. Attack Simulation / Activity

To simulate a brute-force attack, Hydra was executed from the Kali Linux machine targeting the SSH service on the Linux server. Prior to running the automated attack, manual login attempts were performed to verify that authentication logging was functioning correctly and to establish a baseline of normal failed login behavior.

Hydra was then used to generate a high volume of failed login attempts by systematically trying multiple password combinations against the target account. This activity mimics real-world brute-force attacks, where automated tools rapidly attempt credential combinations to gain access.

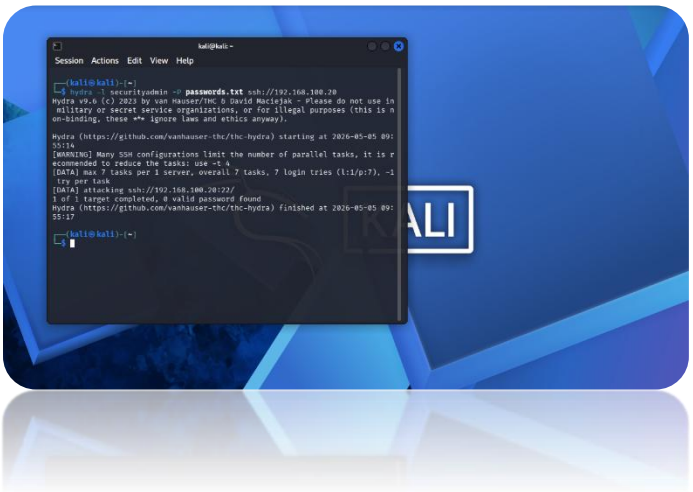
The command used for this activity was:

```
hydra -l securityadmin -P passwords.txt ssh://192.168.100.20
```

This generated repeated authentication failures within a short timeframe, creating the conditions necessary for detection logic to identify abnormal behavior.

This method was chosen because automated tools such as Hydra replicate real-world brute-force attack behavior, generating high-frequency authentication attempts that are necessary for testing detection thresholds and correlation logic.

Figure B2. Hydra brute-force attack execution



6. Detection & Logs

During the attack simulation, the Linux server generated authentication logs that recorded each login attempt, including both successful and failed attempts. These logs were stored in /var/log/auth.log and provided detailed information such as timestamps, usernames, and source IP addresses.

Wazuh successfully ingested these logs through its agent and processed them using its built-in ruleset. Each failed login attempt was identified using rule ID 5503, which detects PAM authentication failures. While this rule provided visibility into individual events, it did not initially identify whether those events were part of a larger attack pattern.

By reviewing both the raw log data and the events displayed in Wazuh, a clear pattern emerged. Multiple failed login attempts originating from the same source IP within a short timeframe indicated automated attack behavior. This observation highlighted the limitation of single-event detection and reinforced the need for correlation-based rules to identify meaningful patterns.

From an analyst perspective, this pattern of repeated failures from a single source IP represents a clear deviation from normal user behavior and is a strong indicator of automated attack activity.

Figure B3. Authentication failure logs on Linux server

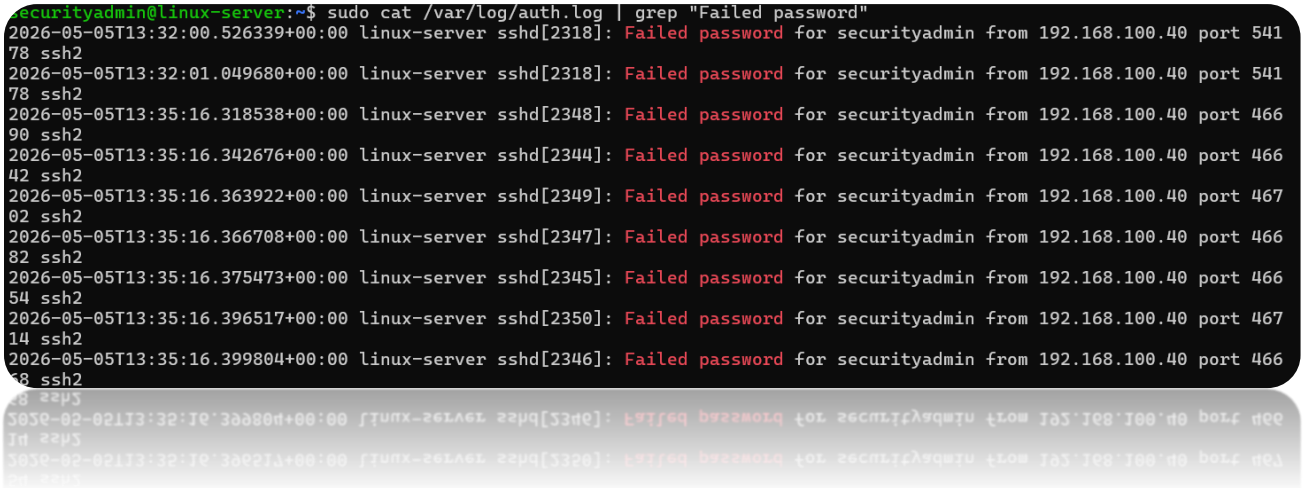
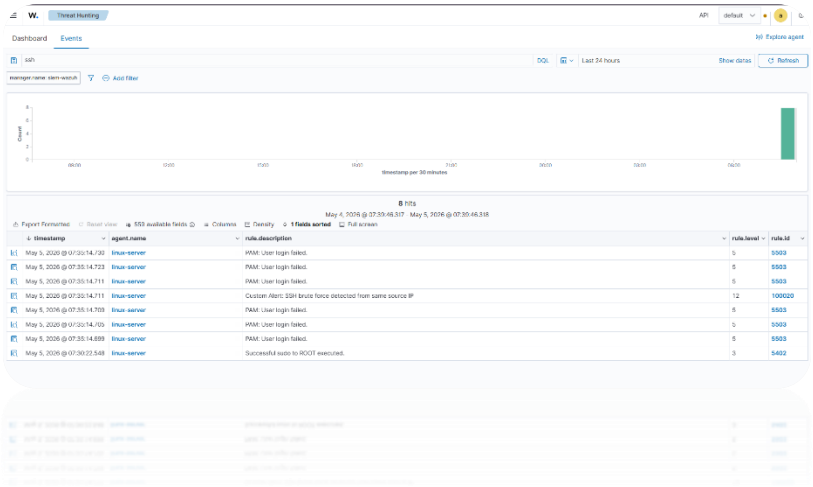


Figure B4. Authentication events displayed in Wazuh SIEM



---

7. Detection Engineering (Layered Rules)

To address the limitations of single-event detection, a layered detection strategy was implemented using custom Wazuh rules. This approach focused on correlating multiple authentication failures into meaningful alerts based on frequency and timing.

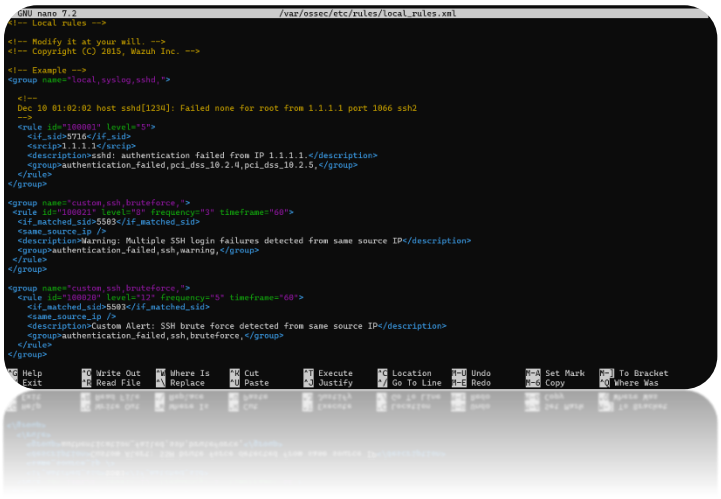
The first rule was designed to act as an early warning mechanism. It triggers when multiple failed login attempts are detected from the same source IP within a short timeframe. This provides visibility into suspicious activity at an early stage, allowing analysts to investigate before the behavior escalates.

The second rule was designed to confirm brute-force activity. It triggers when a higher threshold of failed login attempts is reached within the same timeframe, indicating a strong likelihood of automated attack behavior. This rule generates a higher-severity alert, signaling that the activity has progressed beyond normal user error.

By implementing both rules, the detection system is able to monitor the progression of an attack and provide escalation based on observed behavior. This approach reflects real-world detection engineering practices, where multiple signals are combined to produce more accurate and actionable alerts.

This approach shifts detection from single-event alerting to behavior-based correlation, which is a key principle in modern SIEM detection engineering.

Figure B5. Custom detection rules configuration



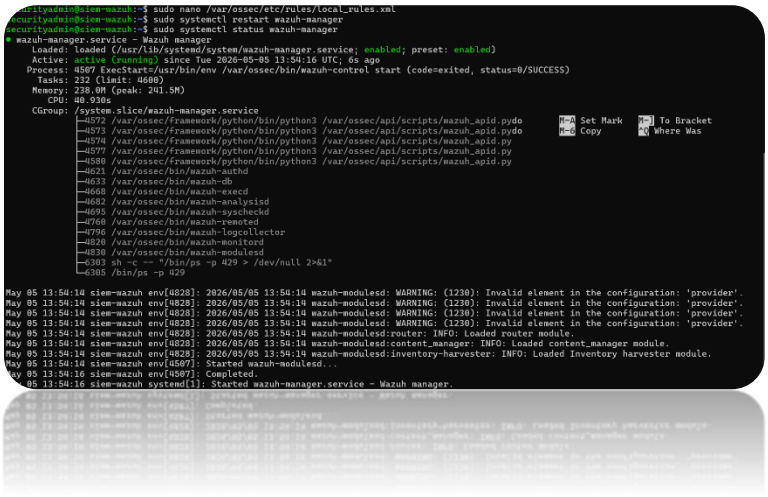
7.5 Challenges & Troubleshooting

During the implementation of custom detection rules, a critical issue was encountered when restarting the Wazuh manager. The service failed to start due to an XML configuration error in the rule file. Specifically, a <rule> element was incorrectly placed outside of a <group> block, which caused the rule parser to fail.

By reviewing system logs and error messages, the issue was identified and resolved by correcting the rule structure and ensuring proper nesting within a group element. Additional troubleshooting involved verifying rule syntax, ensuring correct use of <if_matched_sid>, and confirming that frequency and timeframe attributes were properly configured.

This troubleshooting process was essential in achieving a working detection configuration and provided practical experience in debugging SIEM rule logic, a critical skill in real-world security operations.

Figure B6. Wazuh manager successfully restarted after rule fix



8. Mitigation / Response

In a real-world environment, detection of repeated authentication failures would typically trigger a series of response actions aimed at preventing further unauthorized access. These actions may include temporarily blocking the source IP address, enforcing account lockout policies after a defined number of failed attempts, or implementing rate-limiting controls on authentication services.

Additional defensive measures could involve disabling password-based authentication in favor of SSH key-based authentication, which significantly reduces the risk of brute-force attacks. Continuous monitoring and alerting would also be critical in identifying ongoing attack attempts and responding accordingly.

Although this project focused primarily on detection, it demonstrates how detection capabilities can serve as the foundation for a broader incident response strategy.

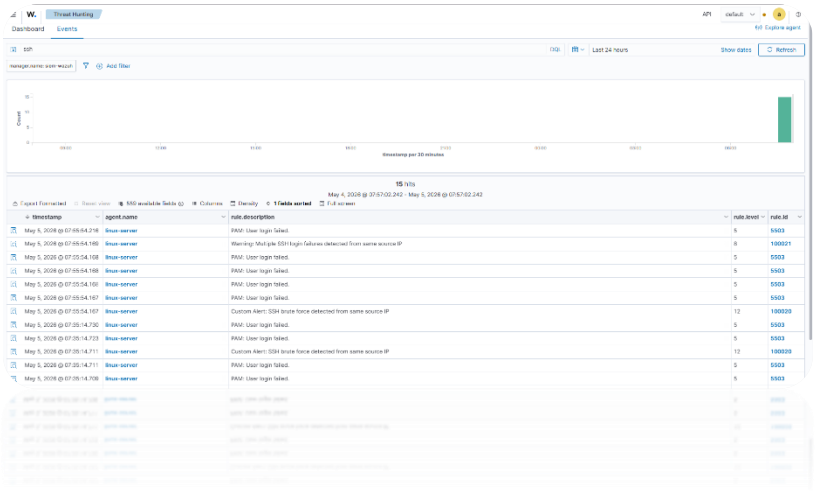
9. Validation

After correcting the rule configuration and implementing the layered detection strategy, the attack simulation was executed again to validate the effectiveness of the detection logic.

The results confirmed that both detection stages were functioning as intended. The early warning rule triggered after a lower threshold of failed attempts, while the high-severity rule triggered after continued attack activity. The alerts accurately reflected the attack pattern and provided clear visibility into the progression of the brute-force attempt. This confirms

that the implemented rules not only function correctly, but also accurately reflect the progression of a brute-force attack in a real-world scenario.

Figure B7. Layered detection alerts (Level 8 and Level 12)



10. Evidence

The evidence collected during this project provides a complete view of the attack and detection lifecycle. This includes Hydra attack execution output, authentication logs generated on the Linux server, and event data captured within the Wazuh SIEM.

Additional evidence includes screenshots of custom rule configurations and alert outputs, which demonstrate how detection logic was implemented and validated. Together, these artifacts support the effectiveness of the detection strategy and provide clear documentation of the entire process from attack simulation to alert generation.

11. Conclusion

This project demonstrates the value of layered detection engineering within a SIEM environment. While default rules provide basic visibility into individual events, they are often insufficient for identifying complex attack patterns such as brute-force activity.

By implementing custom correlation rules and introducing alert escalation logic, it was possible to transform raw log data into meaningful security insights. The layered approach not only improves detection accuracy but also provides earlier visibility into suspicious behavior, enabling more proactive monitoring.

The project also highlights the importance of troubleshooting and validation in SIEM environments. Configuration errors, rule logic issues, and service failures are common challenges that must be addressed to ensure reliable detection. Successfully identifying and resolving these issues provided valuable hands-on experience in real-world detection engineering workflows.

What Could Be Improved

While the detection logic implemented in this project was effective, several improvements could further enhance its capabilities. Expanding detection to include additional related rule IDs and authentication patterns would provide broader coverage.

Additionally, integrating automated response mechanisms such as IP blocking or alert-driven scripts could extend this project beyond detection into active defense. Further tuning of thresholds and timeframes could also help reduce potential false positives while maintaining detection accuracy. This project demonstrates the transition from basic log monitoring to structured detection engineering, emphasizing the importance of correlation, escalation logic, and continuous rule refinement in modern security operations.

---

12. Resume Bullet

Designed and implemented layered Wazuh SIEM detection rules to identify SSH brute-force attacks, correlating authentication failures into early warning and high-severity alerts through simulated attack activity using Hydra.

---

13. MITRE ATT&CK Mapping

This project aligns with the MITRE ATT&CK framework under the **Credential Access** tactic, specifically technique **T1110 – Brute Force**. This technique involves repeated authentication attempts using different credential combinations to gain unauthorized access to a system.

In this lab, brute-force behavior was simulated using Hydra, which generated multiple SSH login attempts against the target system. The detection logic implemented in Wazuh identifies this behavior by correlating repeated authentication failures originating from the same source IP within a defined timeframe.

The layered detection approach enhances this mapping by capturing both early-stage suspicious activity and confirmed attack patterns. The early warning rule detects initial abnormal behavior, while the high-severity rule confirms sustained brute-force activity. This aligns with real-world defensive strategies, where detection systems must identify both indicators of compromise and confirmed attack behavior.

By mapping detection logic to MITRE ATT&CK, this project demonstrates how structured threat frameworks can be used to translate raw log data into meaningful and standardized security insights. This mapping reinforces the importance of aligning detection logic with standardized threat frameworks, enabling consistent analysis, reporting, and threat identification across security environments.

---

14. Future Improvements

Future enhancements to this project could include expanding detection logic to incorporate additional rule IDs, implementing automated response actions such as IP blocking, and integrating broader MITRE ATT&CK coverage.

Additional improvements could involve simulating different attack techniques, refining detection thresholds to reduce false positives, and extending monitoring to additional endpoints to create a more comprehensive detection environment.