Custom Detection Engineering and Incident Analysis Using Wazuh SIEM

Author: Wayne Howlett
Date: May 02, 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 created and validated a custom detection rule that identifies SSH brute-force activity and triggers high-severity alerts.

1. Objective

The goal of this project was to extend a previously built SIEM lab by developing custom detection logic and performing incident analysis. Rather than relying solely on default Wazuh rules, this project focuses on creating a rule that detects brute-force behavior based on repeated authentication failures from a single source. The objective was not only to trigger alerts but also to understand how detection logic works and how events are correlated in a real-world scenario.

---

2. Scenario

In a real-world environment, brute-force attacks are a common method used by attackers to gain unauthorized access. These attacks often involve repeated login attempts over a short period of time, typically targeting services such as SSH.

This project simulates such an attack from a controlled attacker machine (Kali Linux) targeting a Linux server. The SIEM (Wazuh) is configured to monitor authentication logs and detect abnormal login behavior. A custom rule is developed to identify patterns that indicate brute-force activity, mimicking how a SOC analyst or detection engineer would enhance monitoring capabilities in an enterprise environment.

---

3. Environment

The lab environment was built using VirtualBox and consists of multiple virtual machines connected through an isolated internal network. The Wazuh SIEM server was deployed on an Ubuntu system and configured to collect logs from both a Linux server and a Windows endpoint.

The Linux server acts as the primary target for the attack, specifically through SSH authentication. A Kali Linux machine is used as an attacker, running tools designed to simulate brute-force login attempts. All systems communicate over a controlled lab network, ensuring safe and isolated testing conditions.

Figure B1. Lab environment and network configuration

---

4. Tools Used

The primary tool used for detection and monitoring was Wazuh, which provided log ingestion, rule processing, and alert visualization. Kali Linux was used as the attacker machine, specifically leveraging the Hydra tool to simulate brute-force activity. The Linux server generated authentication logs, which were forwarded to Wazuh for analysis.

---

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. The attack attempted multiple password combinations against the root account, generating repeated authentication failures.

The command used for this activity was:

```
hydra -l root -P /usr/share/wordlists/rockyou.txt -t 4 ssh://192.168.100.20
```

This generated a high volume of failed login attempts within a short timeframe, which is typical behavior for brute-force attacks.

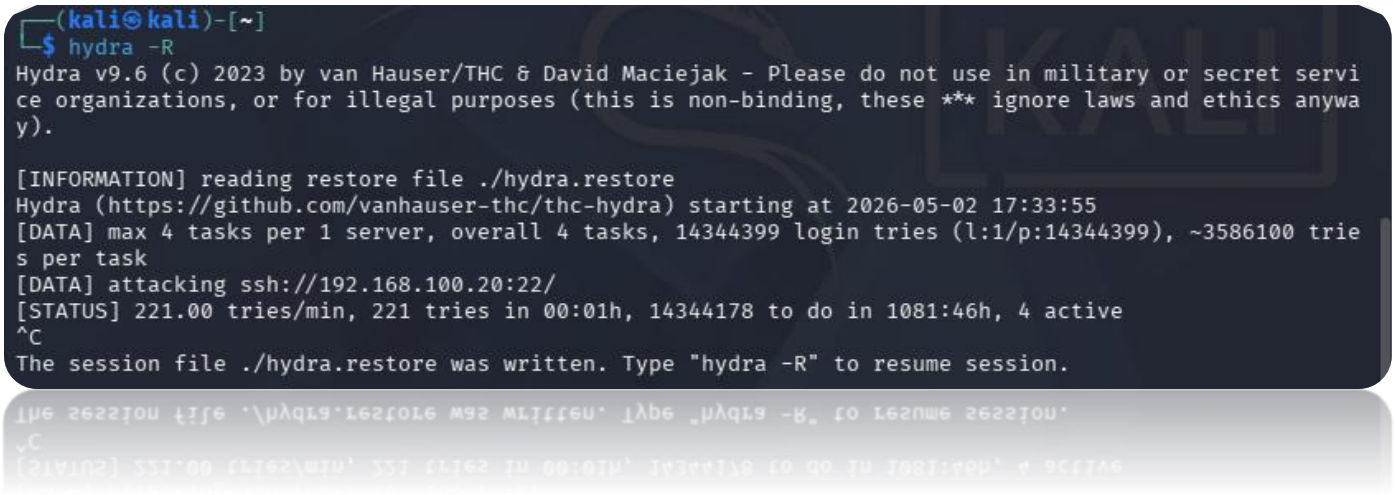

Figure B2. Hydra brute-force attack execution

6. Detection & Logs

Wazuh successfully ingested authentication logs from the Linux server, capturing events such as failed login attempts and repeated password failures. Default rules identified these events individually; however, they did not initially provide a high-confidence detection of brute-force activity.

To improve detection, a custom rule was created to correlate multiple authentication failures from the same source IP within a defined timeframe. This allowed Wazuh to move from simple event logging to behavioral detection.

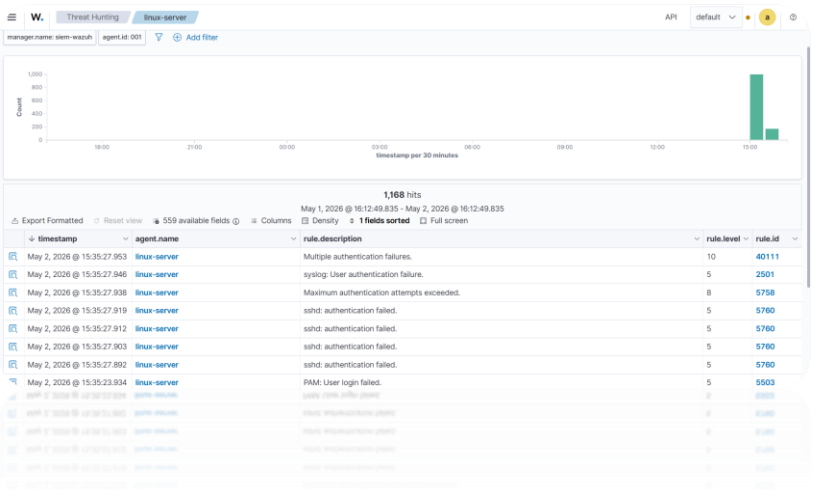

Figure B3. Authentication failure logs in Wazuh

7. Analysis

The attack generated a clear pattern of repeated login failures, all originating from the same source IP address. This behavior is consistent with brute-force attack techniques, where an attacker systematically attempts multiple password combinations.

The custom rule successfully identified this pattern and elevated the severity of the alert to Level 12. This demonstrates how correlating multiple low-level events can provide meaningful detection of malicious activity. The alert also included valuable context such as the source IP, target system, and affected user account, allowing for effective investigation.

7.5 Challenges & Troubleshooting

During the development of this project, several issues were encountered that required troubleshooting and refinement.

One of the primary challenges was that the custom rule did not initially trigger as expected. Despite the presence of multiple authentication failures, no high-severity alerts were generated. This was traced back to incorrect rule syntax, specifically the misuse of `<if_sid>` instead of `<if_matched_sid>` and improper configuration of frequency and timeframe attributes.
Another issue occurred when restarting the Wazuh manager. The service failed due to configuration errors in the rule file. By reviewing system logs and error messages, it was identified that invalid rule formatting and duplicate rule IDs were causing the failure. These issues were resolved by correcting the rule structure and ensuring each rule had a unique identifier.

Additionally, the initial detection logic was too narrow, targeting only a single rule ID. Since SSH authentication failures are generated by multiple rules, the detection approach had to be refined to properly capture relevant events.
These troubleshooting steps were critical in achieving a working detection rule and provided valuable experience in debugging SIEM configurations.

---

8. Mitigation / Response

In a real-world scenario, the detection of brute-force activity would trigger an incident response process. This could include temporarily blocking the source IP, enforcing account lockout policies, or implementing rate-limiting mechanisms on authentication services.

While this lab focused on detection, it demonstrates how such alerts could be used to initiate automated or manual response actions to prevent unauthorized access.

---

9. Validation

After correcting the rule configuration and resolving syntax issues, the attack simulation was executed again. The custom detection rule successfully triggered multiple Level 12 alerts, confirming that the rule was functioning as intended.

The alerts accurately reflected the attack behavior, including repeated authentication failures from a single source IP within a defined timeframe.

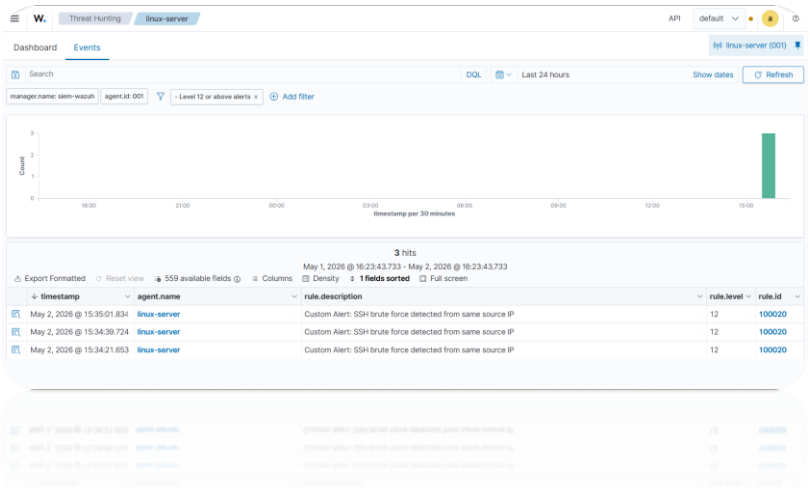

Figure B4. Custom Level 12 alert triggered in Wazuh

---

10. Evidence

The evidence collected during this project includes:

- Hydra attack execution output
- Authentication logs from the Linux server
- Wazuh alert data showing rule correlation
- Detailed alert information including source IP and user account

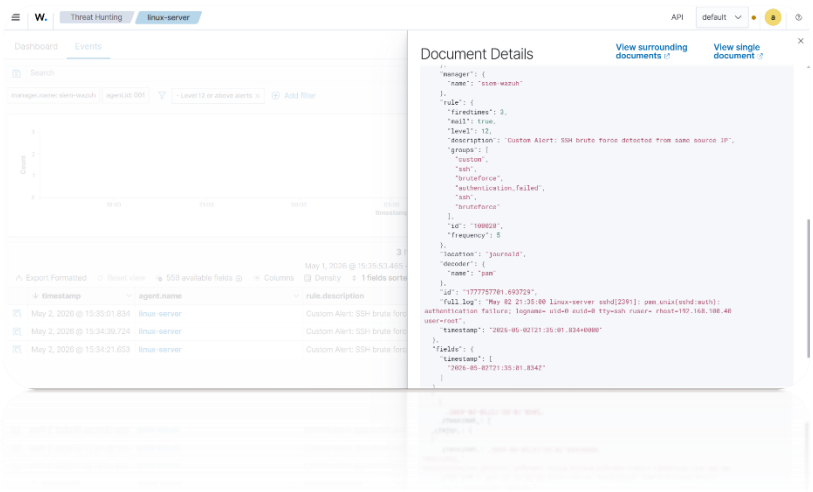

Figure B5. Detailed alert event with source IP and authentication failure

11. Conclusion

This project demonstrates the importance of custom detection engineering in a SIEM environment. While default rules provide baseline visibility, they may not fully capture complex attack patterns. By developing and refining custom rules, it was possible to detect brute-force behavior more effectively.

The project also highlights the importance of troubleshooting and validation, as rule configuration errors can prevent detection from functioning correctly. Overall, this exercise provided hands-on experience in detection logic, log analysis, and incident investigation.

12. Resume Bullet

Designed and implemented a custom Wazuh SIEM detection rule to identify SSH brute-force attacks, correlating multiple authentication failures and triggering high-severity alerts through simulated attack activity using Hydra.

13. MITRE ATT&CK Mapping

T1110 – Brute Force

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 MITRE ATT&CK mapping for broader detection coverage.

Additional attack scenarios and endpoint monitoring could also be introduced to further strengthen the lab environment.

Appendix B – Evidence