SOC Automation & Web Application Defense

Date: January 2, 2026

Author: Virat Solanki

Subject: Implementation of WAF and SIEM for Automated Threat Detection

---

1. Executive Summary

This project demonstrates the design and implementation of a Security Operations Center (SOC) workflow to detect and mitigate web-based cyber threats. The objective was to secure a vulnerable web server against OWASP Top 10 attacks, specifically SQL Injection (SQLi).

The architecture integrates **ModSecurity (Web Application Firewall)** for network-level filtering and **Wazuh (SIEM)** for centralized log analysis and alerting. The project successfully demonstrates the "Attack-Detect-Defend" lifecycle, culminating in the automated blocking of malicious traffic.

2. Lab Environment & Architecture

2.1 Infrastructure

- **Victim Server:** Ubuntu Linux running Apache HTTP Server and MySQL.
- **Vulnerable Application:** DVWA (Damn Vulnerable Web App) configured with legacy vulnerabilities.
- **Security Tools:**
    - **Wazuh Manager:** Centralized SIEM for log correlation.
    - **Wazuh Agent:** Endpoint monitoring and log forwarding.
    - **ModSecurity:** Open-source WAF with OWASP Core Rule Set (CRS).
- **Attacker Machine:** Windows 10 utilizing browser-based payload injection.

---

3. Methodology & Implementation

Phase 1: Vulnerability Assessment (Red Team)

The initial phase involved identifying security gaps in the target application. I deployed the DVWA platform and performed a manual penetration test using SQL Injection techniques.

- **Attack Vector:** Input fields vulnerable to unsanitized queries.
- **Payload Used:** `' OR '1'='1` (Classic Boolean-based SQLi).
- **Impact:** Successful bypass of authentication, resulting in full disclosure of the backend database user list.

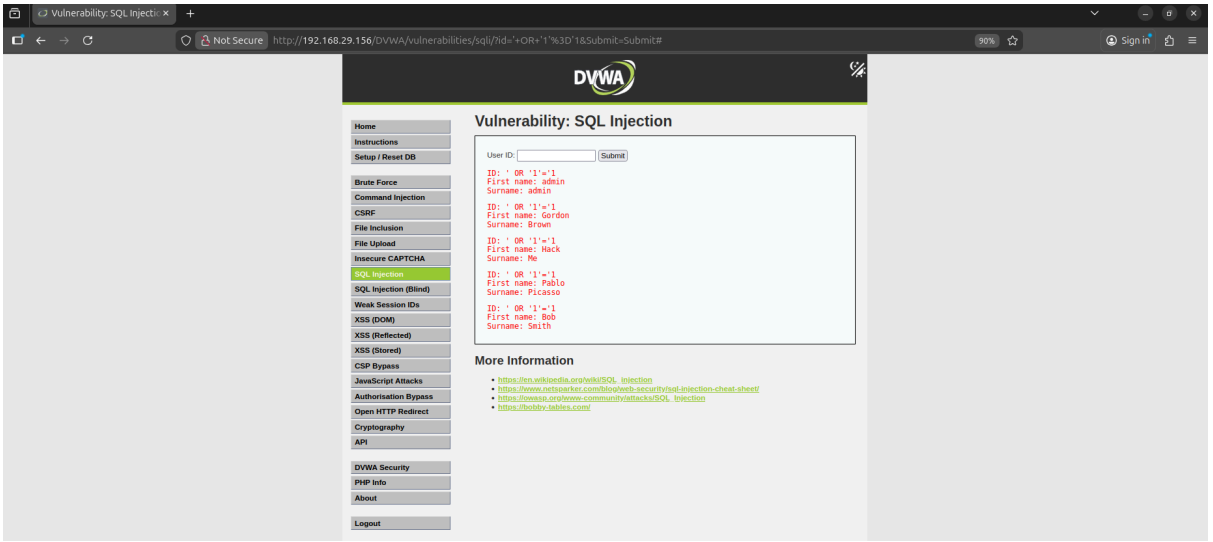

Figure 1: Successful exploitation of SQL Injection vulnerability.

Phase 2: SIEM Rule Development (Blue Team)

To monitor the attack surface, I integrated ModSecurity logs (`/var/log/apache2/error.log`) with the Wazuh Agent. A gap analysis of the default SIEM rules revealed that WAF warnings were not triggering high-priority alerts.

I developed a **Custom Detection Rule (ID 100100)** in XML to parse specific threat patterns and elevate them to Critical severity.

Custom Rule Logic:

```
XML

<rule id="100100" level="12">
  <if_sid>30401</if_sid>
  <match>SQL Injection</match>
  <description>WAF Critical: SQL Injection Attack Detected</description>
  <group>modsecurity,attack,</group>
</rule>
```

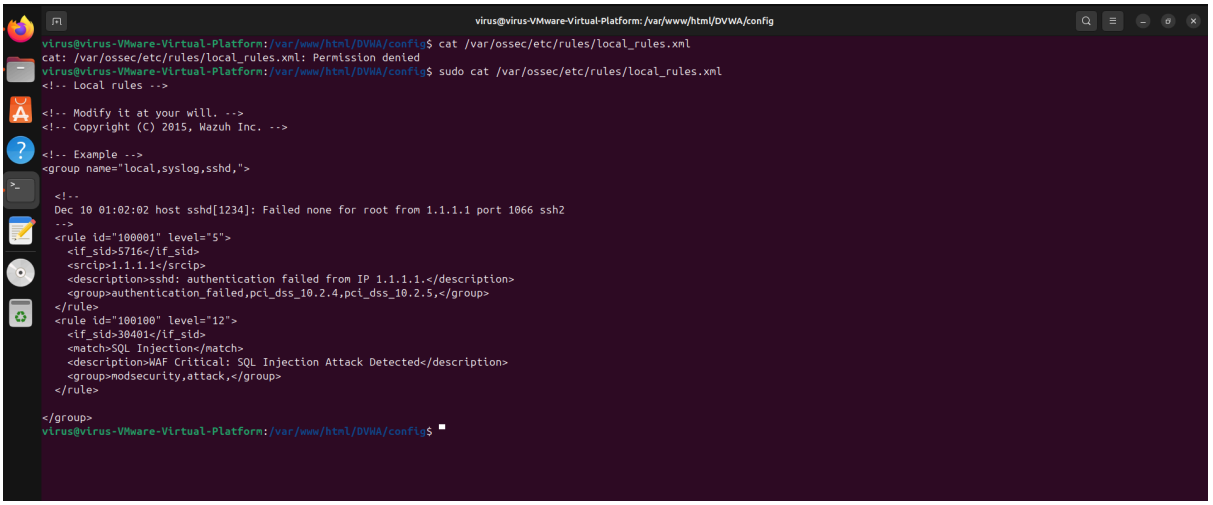

Figure 2: Configuration of custom SIEM rule in local_rules.xml.

Phase 3: Threat Detection & Analysis

With the custom rules active, I re-executed the attack payload. The Wazuh Manager successfully ingested the raw logs from the Ubuntu server, parsed the ModSecurity event, and correlated it against the custom rule.

The SIEM dashboard triggered a **Level 12 Security Alert**, providing real-time visibility into the attacker's IP address, the timestamp, and the specific attack signature ("SQL Injection").

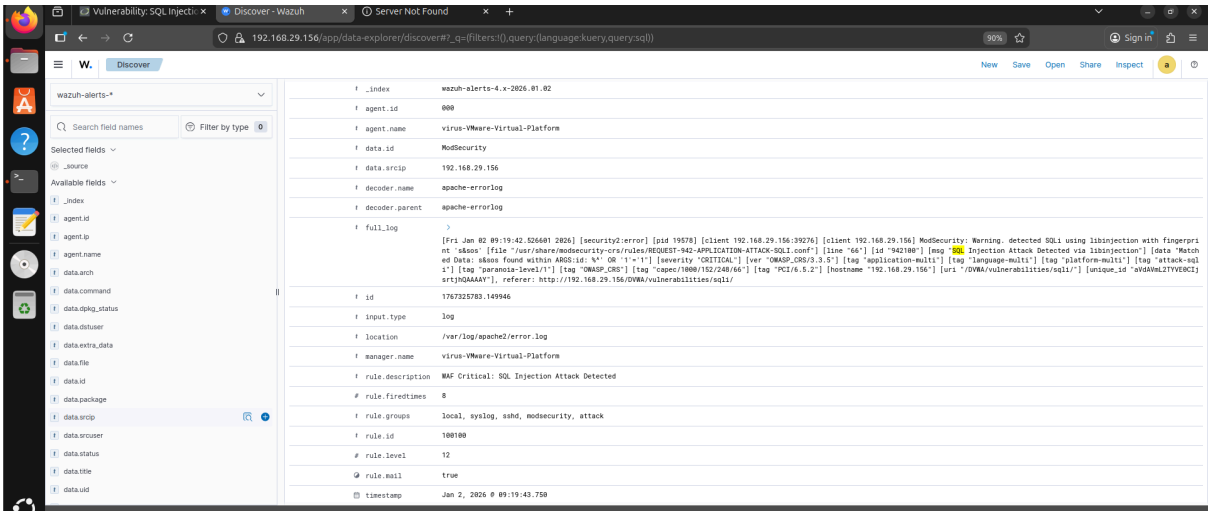

Figure 3: Wazuh Dashboard capturing the attack in real-time.

Phase 4: Mitigation & Hardening

The final phase focused on remediation. I transitioned the ModSecurity engine from "Detection Only" mode to "Blocking" mode by modifying the `/etc/modsecurity/modsecurity.conf` configuration file.

Upon re-attempting the SQL Injection attack, the WAF intercepted the malicious HTTP request and terminated the session immediately, returning a **403 Forbidden** error code. This effectively patched the vulnerability without requiring changes to the application source code.

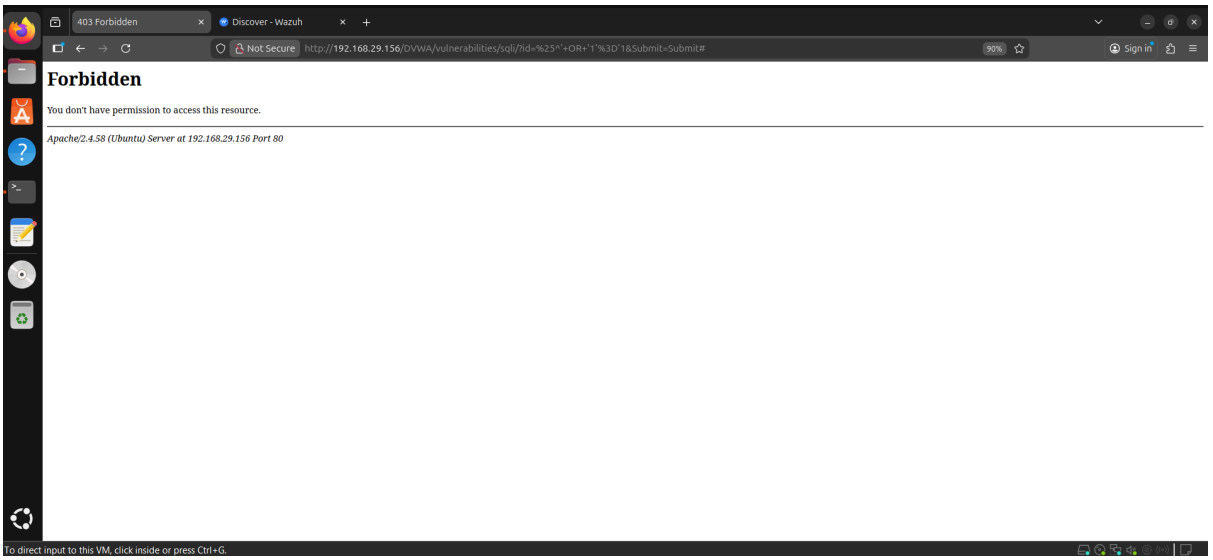

Figure 4: Successful blocking of the attack payload by the WAF.

---

4. Conclusion

This project validates the effectiveness of a layered defense strategy. By combining a Web Application Firewall for immediate threat blocking with a SIEM for long-term analysis and alerting, organizations can significantly reduce their exposure to web-based attacks. The implementation proves that legacy applications can be secured externally through proper SOC tooling and configuration.