# Defense-in-Depth Security Lab: NGFW, WAF & SIEM Implementation

Prepared by: Saboor Ali

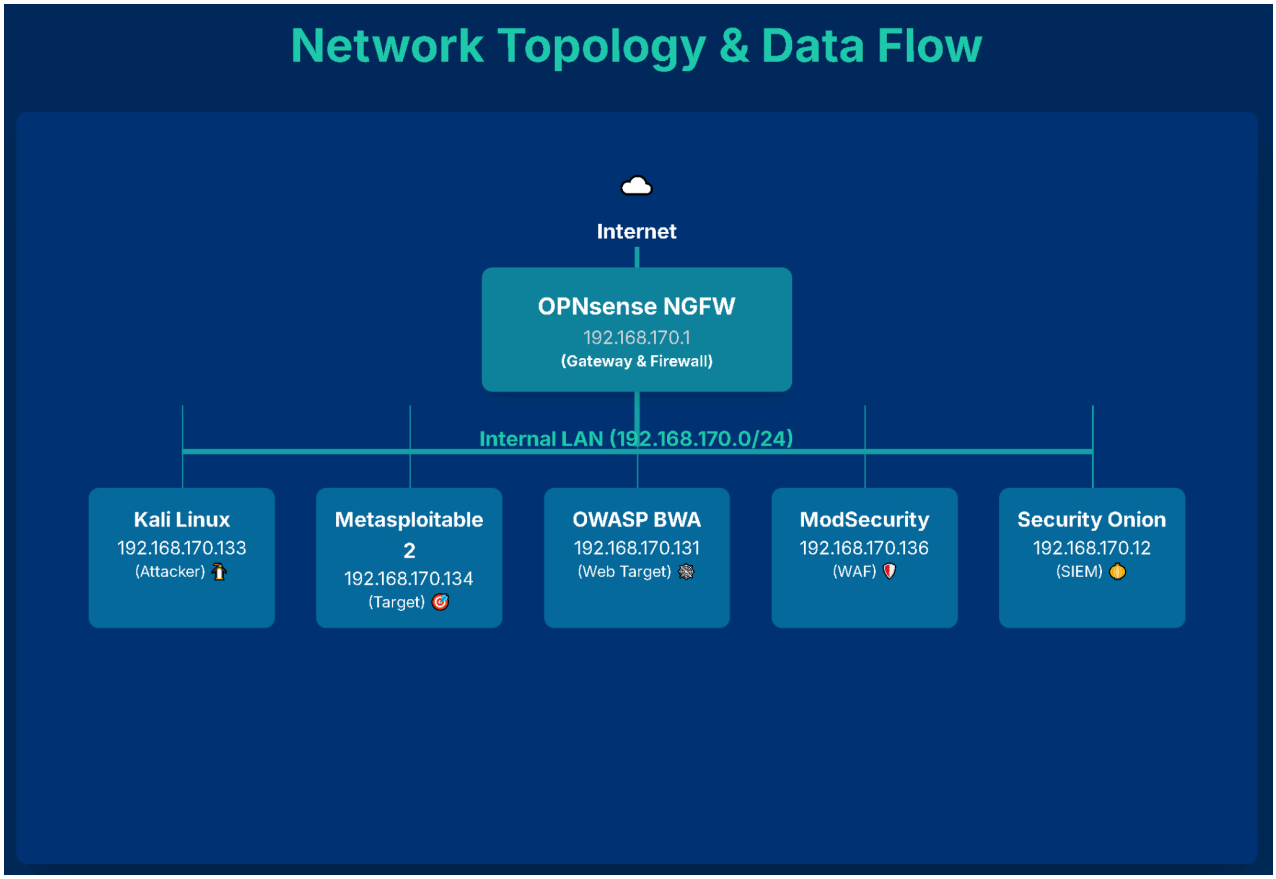
Date: September 27, 2025

## Overview

This lab replicates a real-world enterprise environment to illustrate a defense-in-depth strategy combining a Next-Generation Firewall (NGFW), Web Application Firewall (WAF), and Security Information and Event Management (SIEM) to counter both network-level and web application attacks.

## Lab Environment Setup

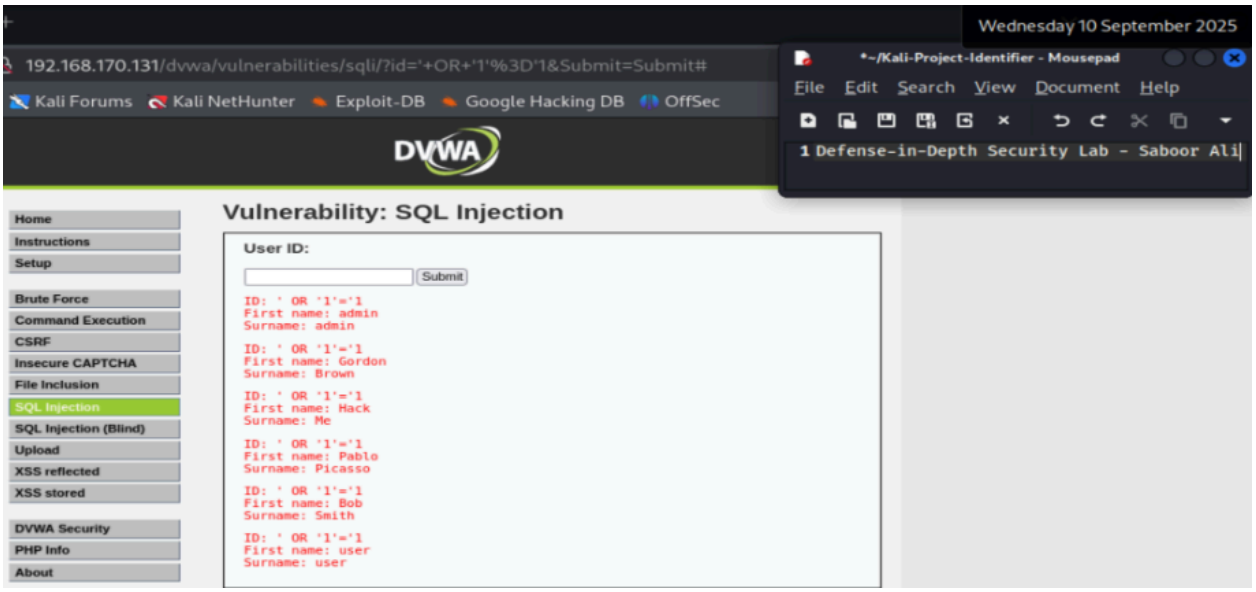
VMs include Kali Linux (attacker), Metasploitable 2, OWASP BWA/DVWA (vulnerable targets), OPNsense (NGFW), Security Onion (SIEM) and ModSecurity WAF. All IP assignments used DHCP; IPs vary in attack logs due to later changes—static IP steps are omitted as not essential to defense outcomes.



# Baseline Vulnerability Testing (Pre-defense)

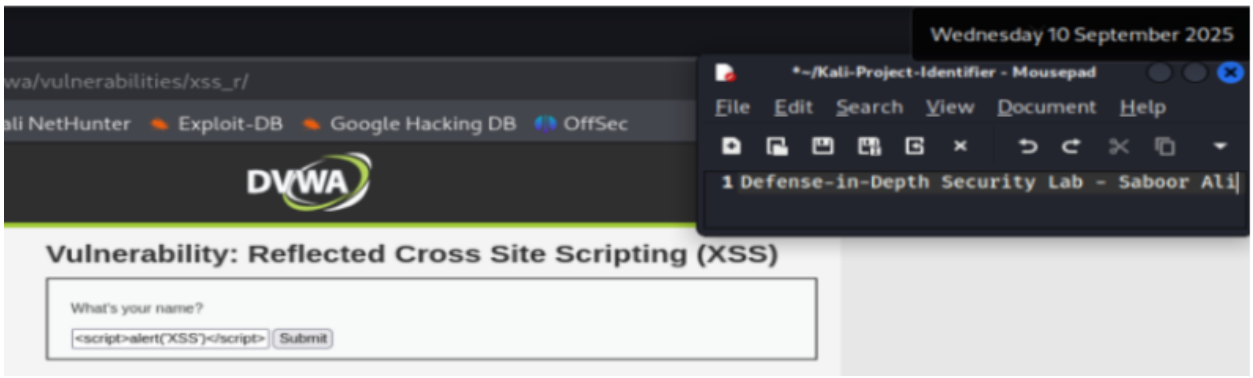
## Vulnerability: SQL Injection

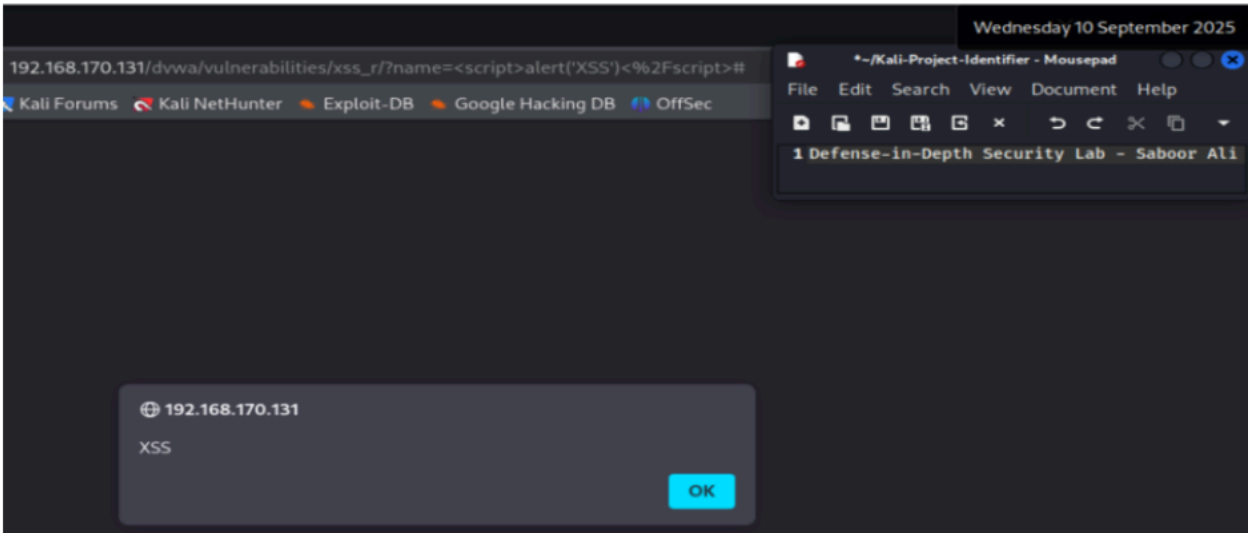
- A SQL Injection attack using the OR 1=1 payload against DVWA successfully bypassed authentication controls, resulting in unauthorized access and data exfiltration.


## Vulnerability: Cross-Site Scripting (XSS)

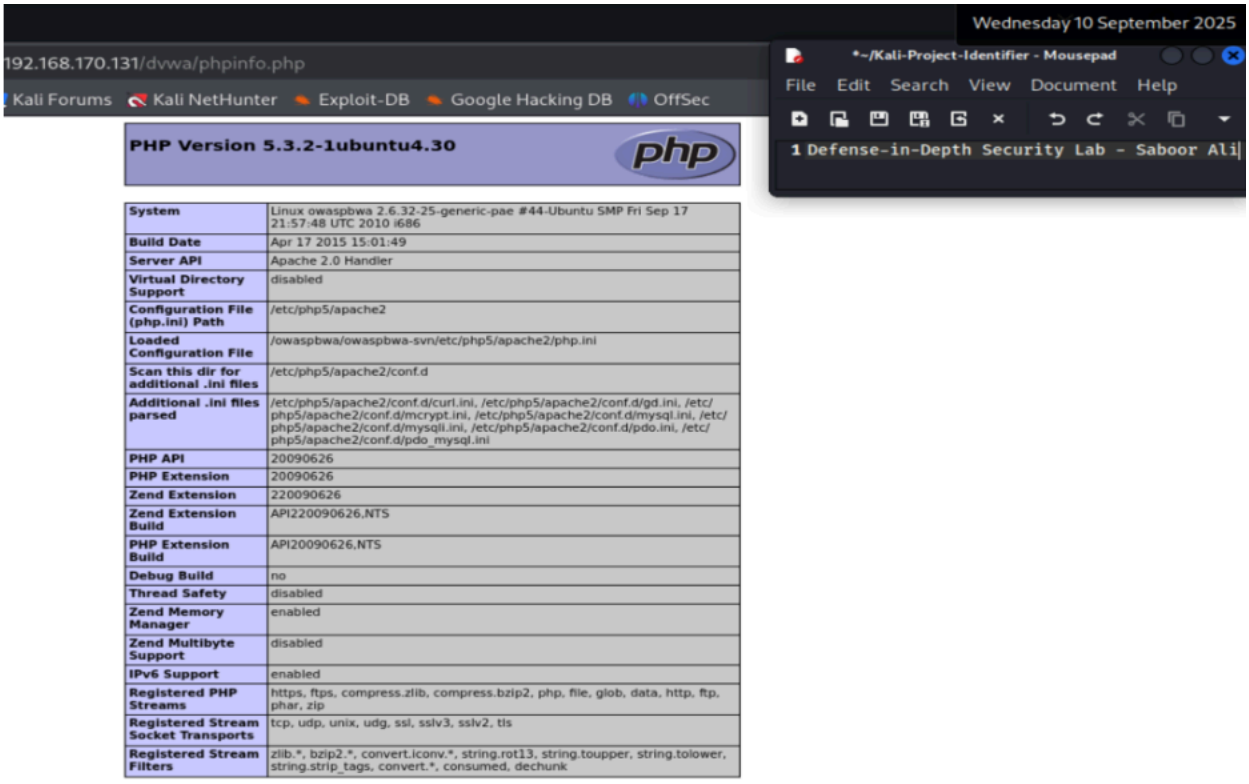
- <script>alert('XSS')</script> submitted to DVWA XSS module, browser displays pop-up.






## Vulnerability: Directory Traversal and Information Exposure

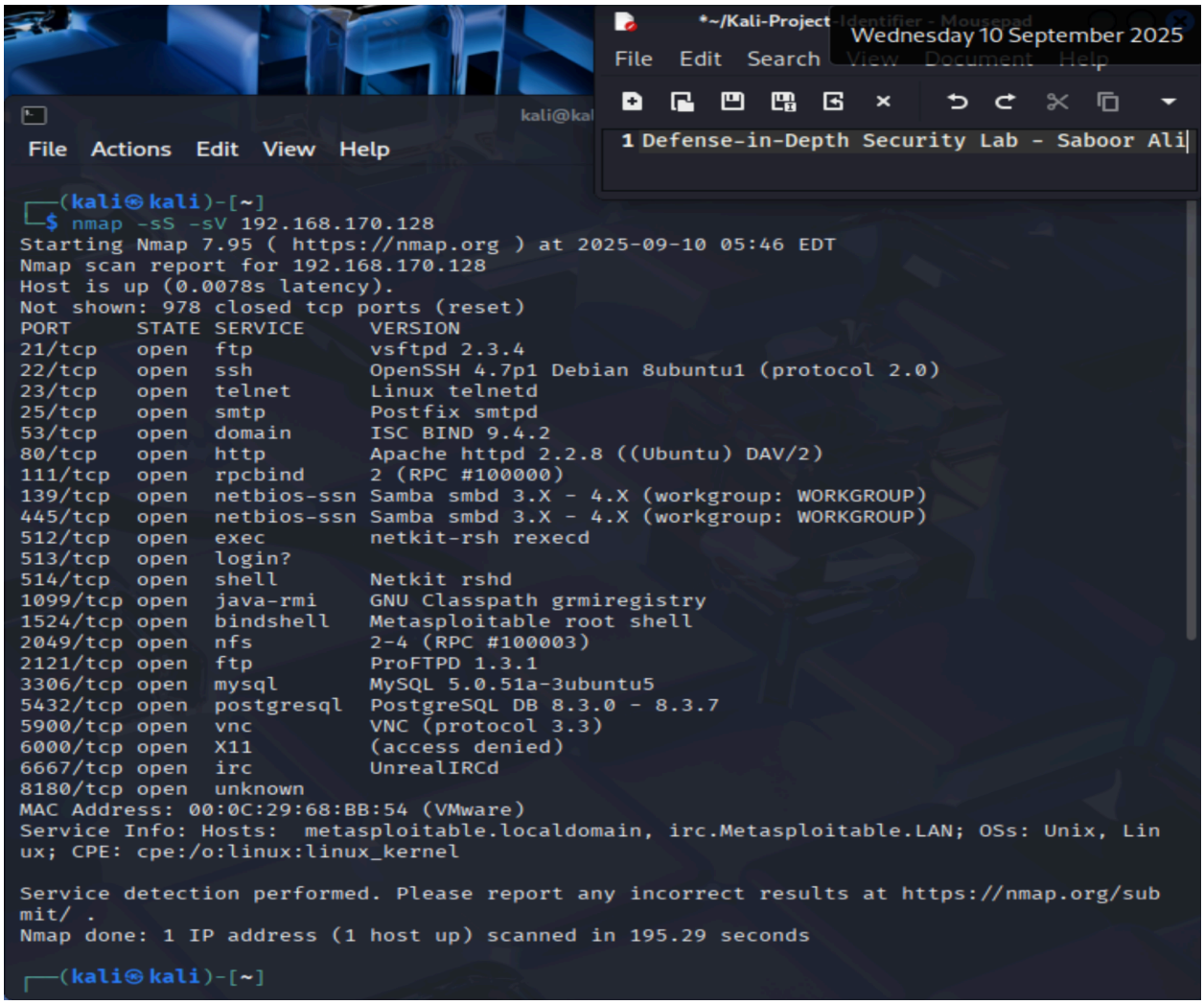
- Direct access to /phpinfo.php, /test/ confirms exposures.




| | |
|---|---|
| System | Linux owaspbwa 2.6.32-25-generic-pae #44-Ubuntu SMP Fri Sep 17 21:57:48 UTC 2010 i686 |
| Build Date | Apr 17 2015 15:01:49 |
| Server API | Apache 2.0 Handler |
| Virtual Directory Support | disabled |
| Configuration File (php.ini) Path | /etc/php5/apache2 |
| Loaded Configuration File | /owaspbwa/owaspbwa-svn/etc/php5/apache2/php.ini |
| Scan this dir for additional .ini files | /etc/php5/apache2/conf.d |
| Additional .ini files parsed | /etc/php5/apache2/conf.d/curl.ini, /etc/php5/apache2/conf.d/gd.ini, /etc/php5/apache2/conf.d/mcrypt.ini, /etc/php5/apache2/conf.d/mysql.ini, /etc/php5/apache2/conf.d/mysqli.ini, /etc/php5/apache2/conf.d/pdo.ini, /etc/php5/apache2/conf.d/pdo_mysql.ini |
| PHP API | 20090626 |
| PHP Extension | 20090626 |
| Zend Extension | 220090626 |
| Zend Extension Build | API220090626,NTS |
| PHP Extension Build | API20090626,NTS |
| Debug Build | no |
| Thread Safety | disabled |
| Zend Memory Manager | enabled |
| Zend Multibyte Support | disabled |
| IPv6 Support | enabled |
| Registered PHP Streams | https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip |
| Registered Stream Socket Transports | tcp, udp, unix, udg, ssl, sslv3, sslv2, tls |
| Registered Stream Filters | zlib.*, bzip2.*, convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk |

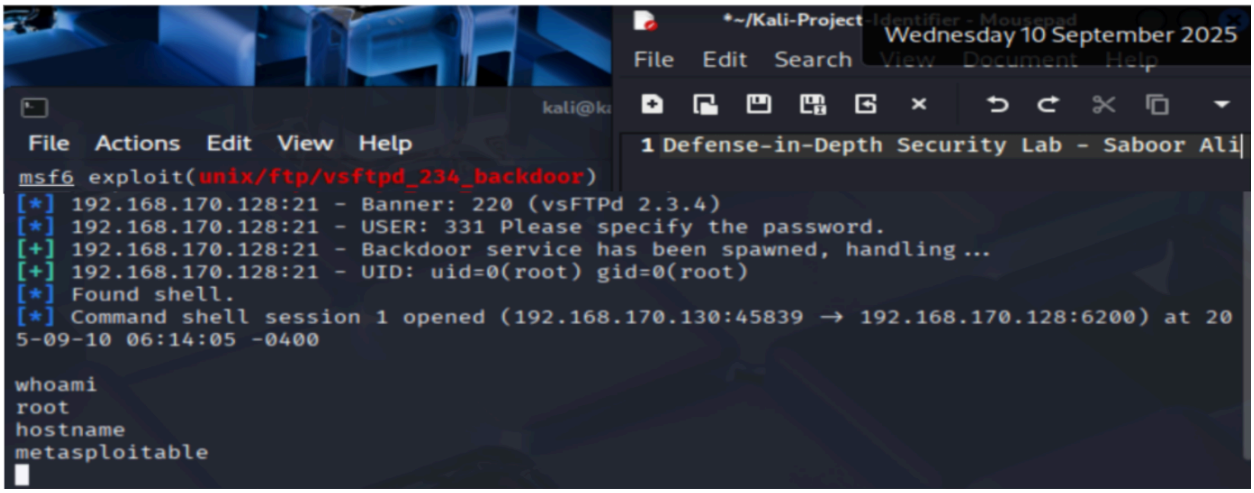
## Vulnerability: Network Enumeration

- Nmap scan uncovers open services and banners on Metasploitable 2.




### Vulnerability: FTP Backdoor Exploit (Metasploitable)

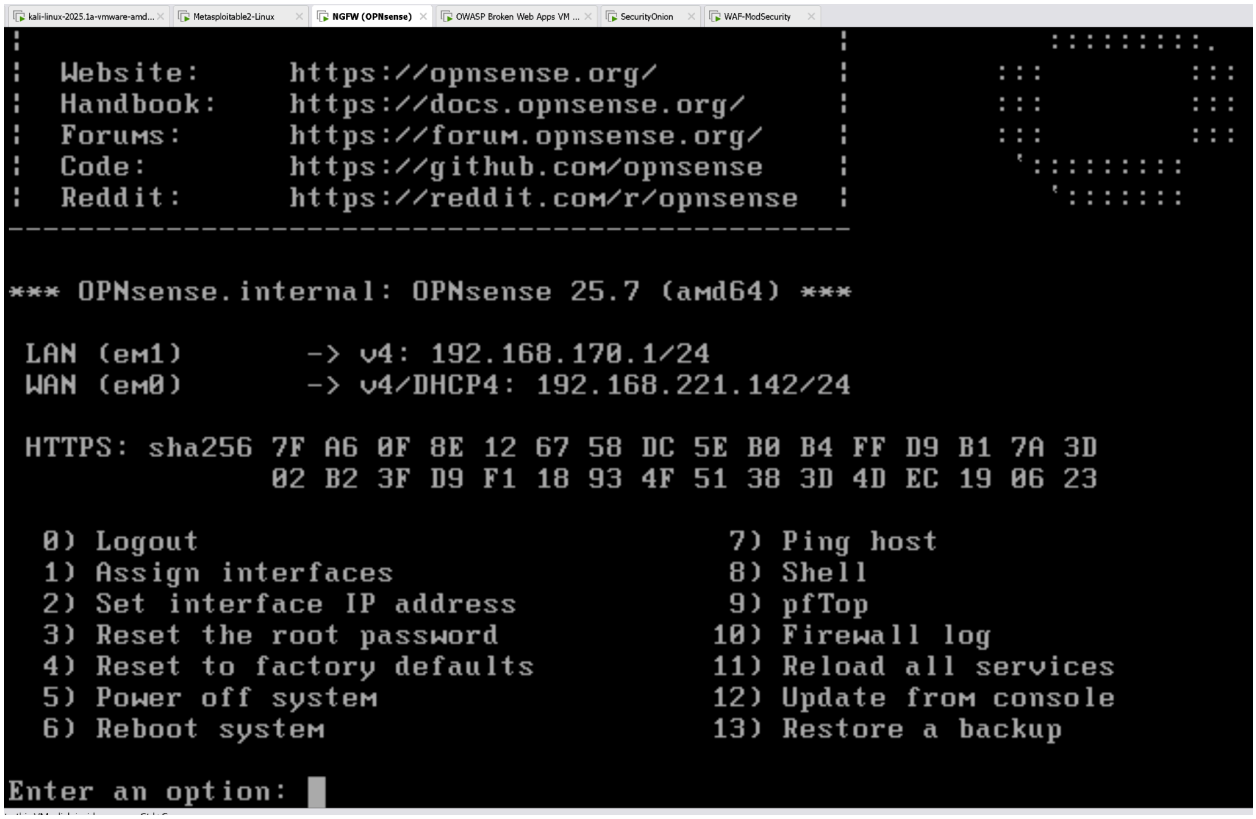
- Metasploit is used to execute a backdoor exploit against Metasploitable, successfully opening a shell.

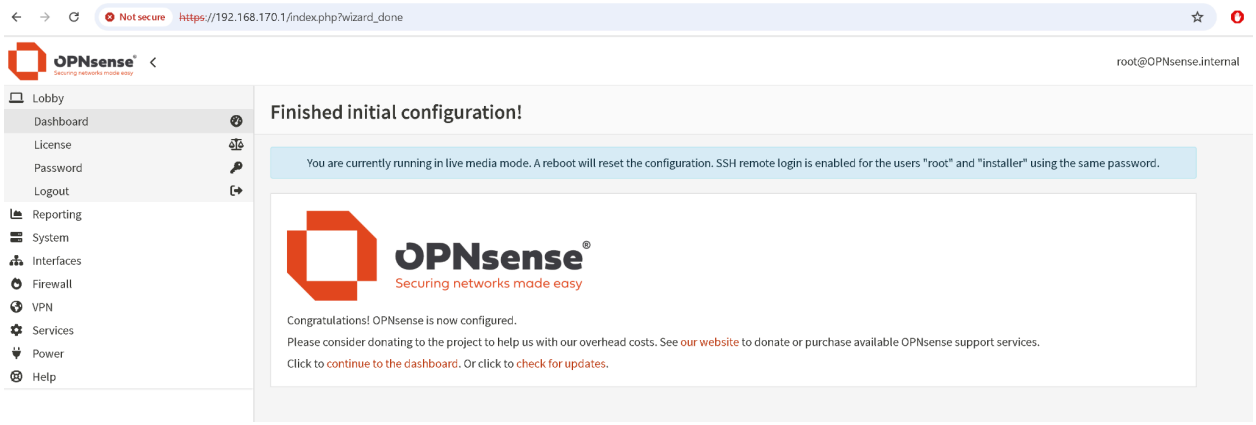


# Security Controls Deployment

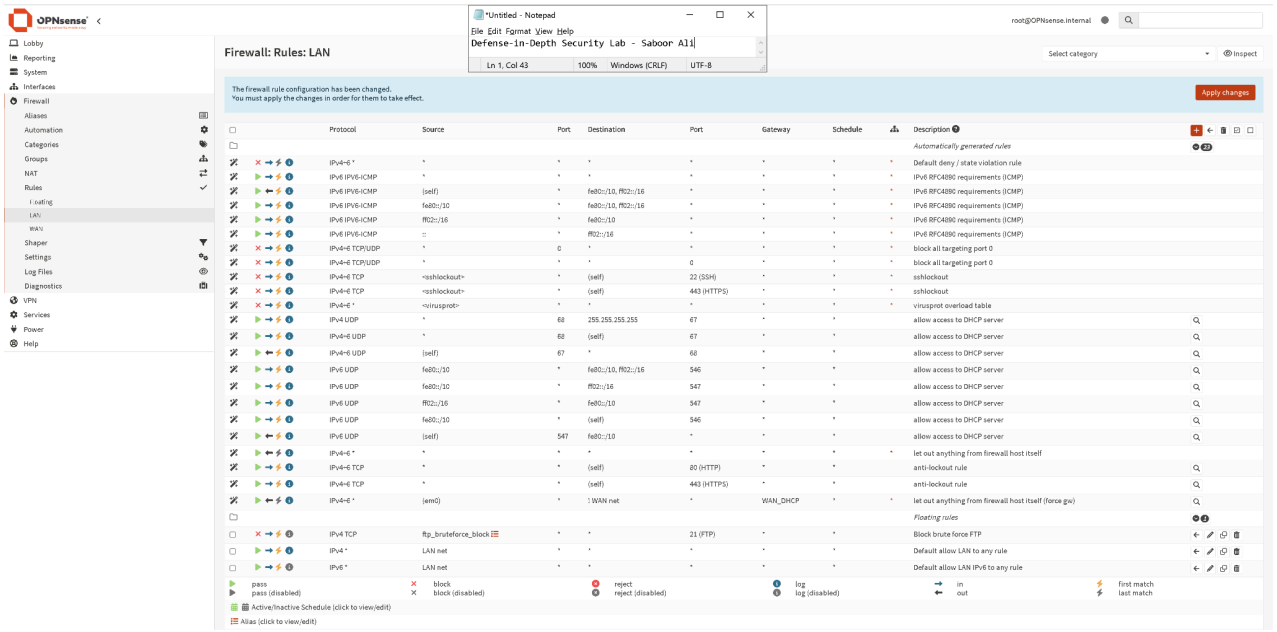
## 1. NGFW (OPNsense) Setup

- OPNsense VM is installed and configured with dual interfaces. Web UI confirms system status.



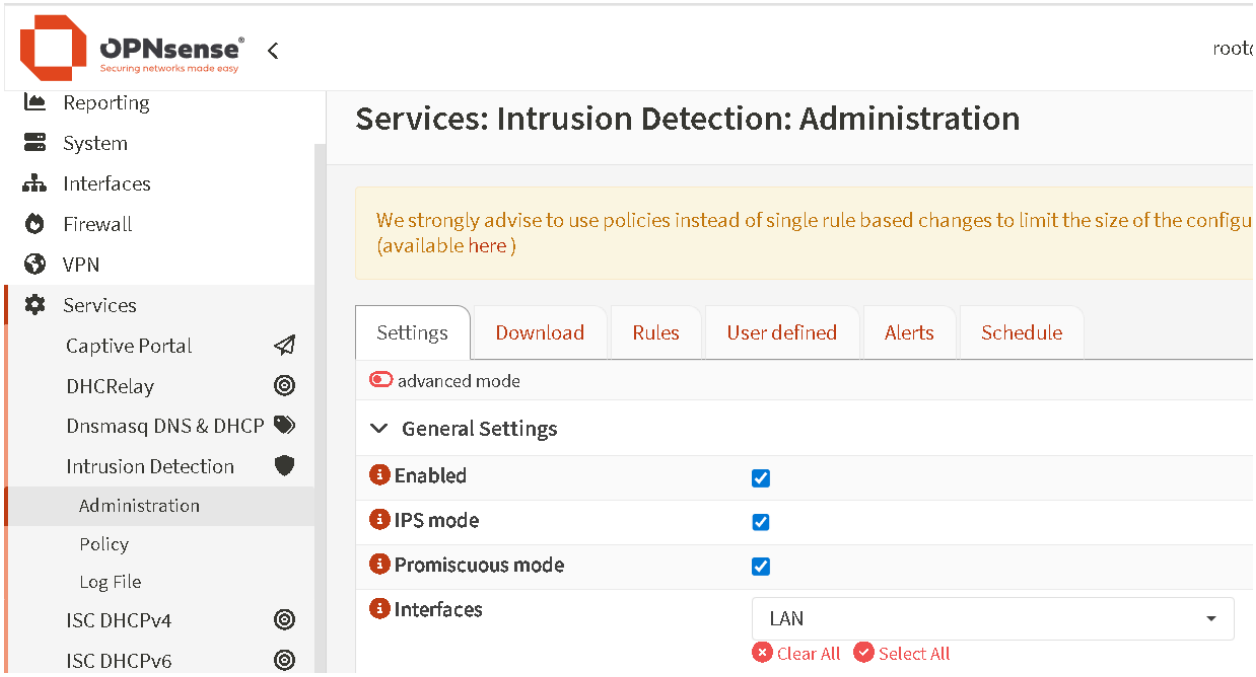



- **LAN rules:** Allow only HTTP(S), SSH for management; FTP restricted—Brute-force countermeasures implemented with address block alias and connection thresholds.

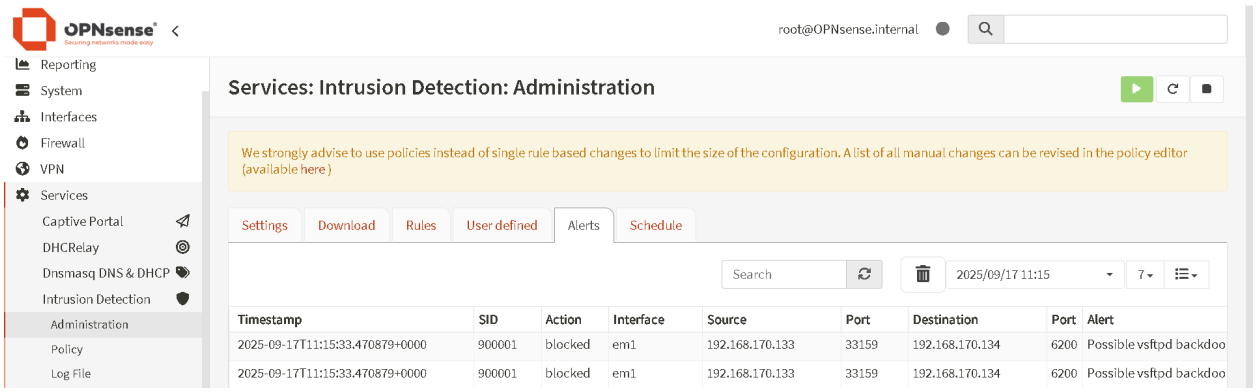



.

- **Intrusion Prevention and Detection:** Suricata was activated within the OPNsense dashboard to provide real-time alerting and proactive blocking of known malicious signatures.




- **Mitigation Example:** A detected vsFTPd backdoor attack is promptly blocked by a custom drop rule.




---

## 2. Web Application Firewall (WAF) ModSecurity

- Configured for Nginx reverse proxy in front of DVWA/OWASP BWA targets.

```
  GNU nano 6.2                                        nginx.conf *
    server {
        listen       80;
        server_name  192.168.170.136;

        root /usr/local/nginx/html;


       # Enable ModSecurity

       modsecurity on;

       modsecurity_rules_file /usr/local/nginx/conf/modsecurity.conf;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location / {

        proxy_pass http://192.168.170.131;

        proxy_set_header Host $host;

        proxy_set_header X-Real-IP $remote_addr;

        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

        }_
```

- **Rules Activated:** OWASP CRS rules loaded (detect/block modes).

```
Home | NGFW (OPNsense) | OWASP Broken Web Apps VM ... | Metasploitable2-Linux | kali-linux-2025.1a-vmware-amd... | SecurityOnion | WAF-ModSecurity
  GNU nano 6.2                                        modsecurity.conf *
# Enable ModSecurity

SecRuleEngine On

SecRequestBodyAccess On

SecAuditEngine On

SecResponseBodyAccess Off

SecDataDir /usr/local/modsecurity/data

SecTmpDir /usr/local/modsecurity/tmp

SecAuditLog /usr/local/nginx/logs/modsec_audit.log

SecAuditLogParts ABIJDEFHZ

SecDebugLog /usr/local/nginx/logs/modsec_debug.log

SecDebugLogLevel 3

# Include CRS setup

Include /usr/local/modsecurity-crs/crs-setup.conf

# Include all CRS rules

Include /usr/local/modsecurity-crs/rules/*.conf
```

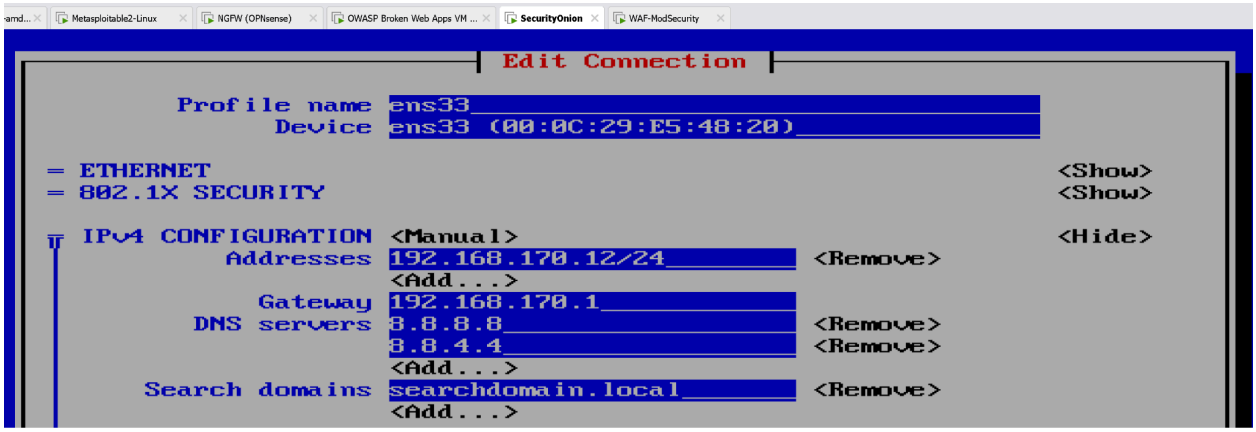
- **Example Mitigation:** SQL Injection attempts receive HTTP 403; detection logged.

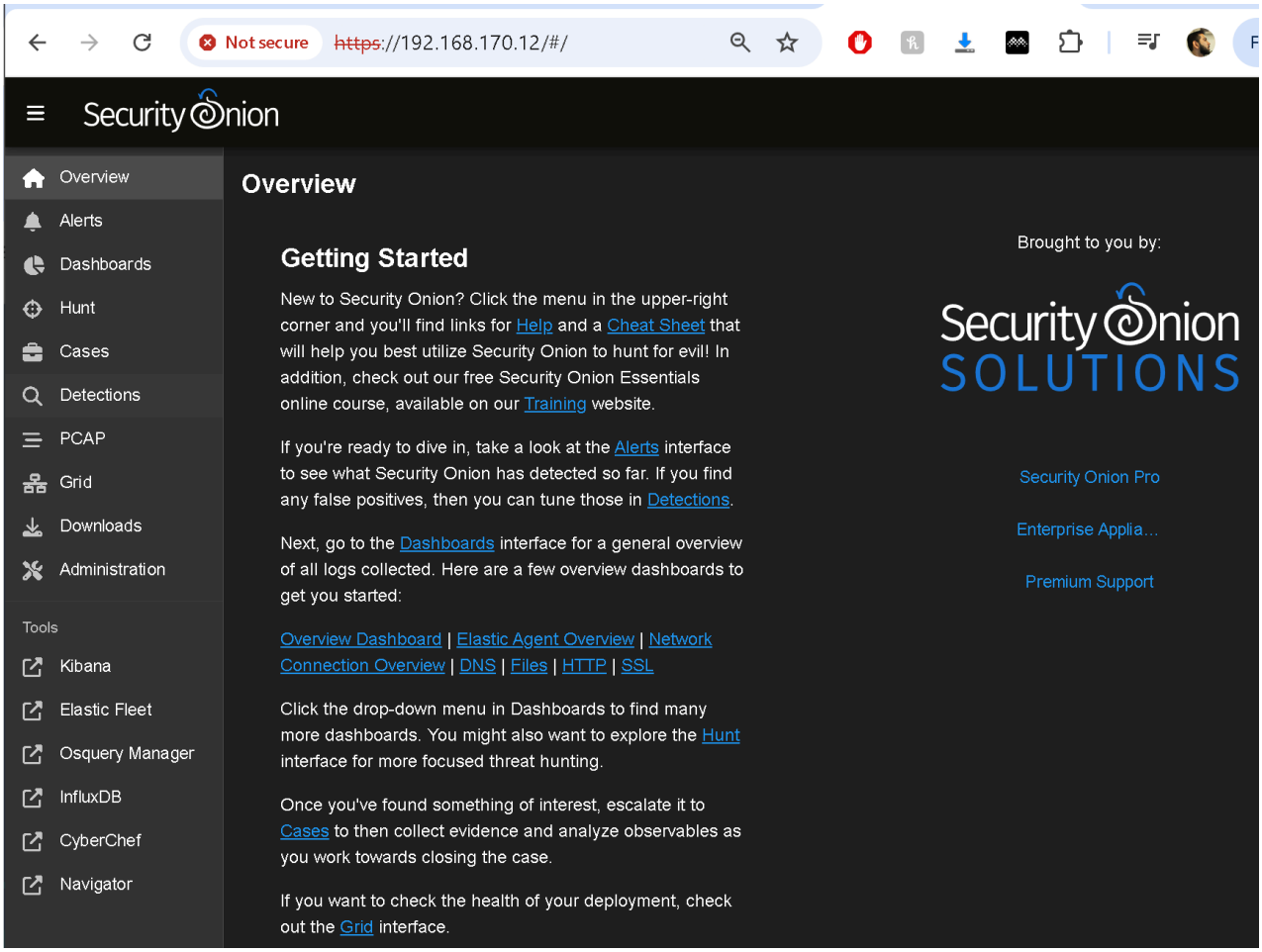
```
waf@waf-modsecurity:/usr/local/nginx/logs$ tail modsec_audit.log
ModSecurity: Warning. Matched "Operator `Rx' with parameter `(?:^([\d.]+|\[[\da-f:]+\]|[\da-f:]+)(:[
\d]+)?$)' against variable `REQUEST_HEADERS:Host' (Value: `192.168.170.136' ) [file "/usr/local/mods
ecurity-crs/rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf"] [line "711"] [id "920350"] [rev ""] [msg "
Host header is a numeric IP address"] [data "192.168.170.136"] [severity "4"] [ver "OWASP_CRS/4.19.0
-dev"] [maturity "0"] [accuracy "0"] [tag "application-multi"] [tag "language-multi"] [tag "platform
-multi"] [tag "attack-protocol"] [tag "paranoia-level/1"] [tag "OWASP_CRS"] [tag "OWASP_CRS/PROTOCOL
-ENFORCEMENT"] [tag "capec/1000/210/272"] [hostname "192.168.170.136"] [uri "/dvwa/vulnerabilities/s
qli/"] [unique_id "175764488172.546717"] [ref "o0,15o0,15v86,15"]
ModSecurity: Warning. detected SQLi using libinjection. [file "/usr/local/modsecurity-crs/rules/REQU
EST-942-APPLICATION-ATTACK-SQLI.conf"] [line "46"] [id "942100"] [rev ""] [msg "SQL Injection Attack
 Detected via libinjection"] [data "Matched Data: s&sos found within ARGS:id: ' OR '1'='1"] [severit
y "2"] [ver "OWASP_CRS/4.19.0-dev"] [maturity "0"] [accuracy "0"] [tag "application-multi"] [tag "la
nguage-multi"] [tag "platform-multi"] [tag "attack-sqli"] [tag "paranoia-level/1"] [tag "OWASP_CRS"]
 [tag "OWASP_CRS/ATTACK-SQLI"] [tag "capec/1000/152/248/66"] [hostname "192.168.170.136"] [uri "/dvw
a/vulnerabilities/sqli/"] [unique_id "175764488172.546717"] [ref "v35,11"]
ModSecurity: Access denied with code 403 (phase 2). Matched "Operator `Ge' with parameter `5' agains
t variable `TX:BLOCKING_INBOUND_ANOMALY_SCORE' (Value: `8' ) [file "/usr/local/modsecurity-crs/rules
/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "222"] [id "949110"] [rev ""] [msg "Inbound Anomaly Sc
ore Exceeded (Total Score: 8)"] [data ""] [severity "0"] [ver "OWASP_CRS/4.19.0-dev"] [maturity "0"]
 [accuracy "0"] [tag "anomaly-evaluation"] [tag "OWASP_CRS"] [hostname "192.168.170.136"] [uri "/dvw
a/vulnerabilities/sqli/"] [unique_id "175764488172.546717"] [ref ""]
```

---

## 3. SIEM (Security Onion/Kibana OSSEC Integration)

- Security Onion setup steps, log collection configuration, and dashboard navigation are shown.

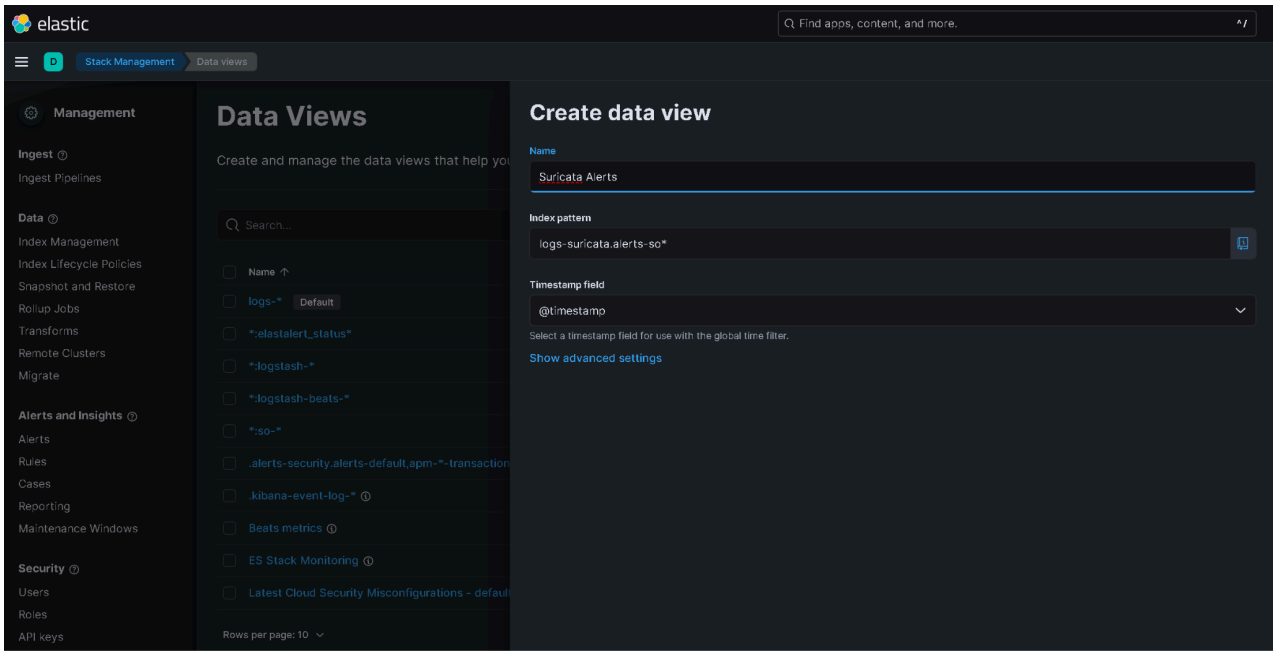




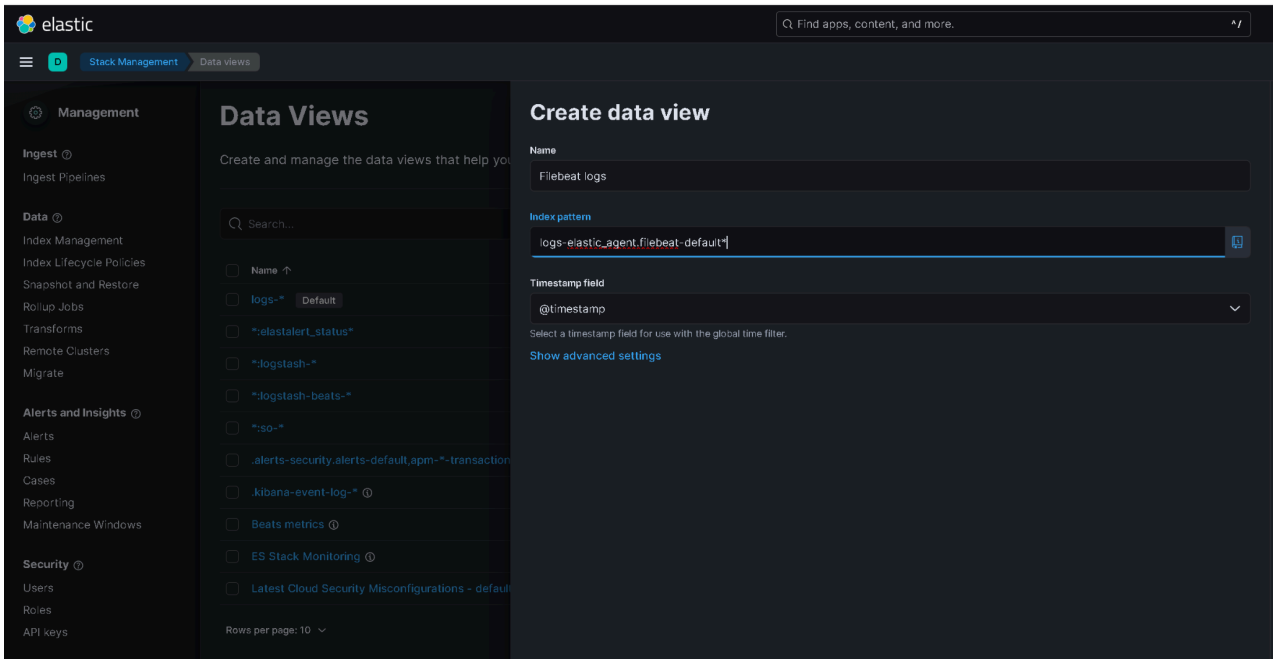

## Kibana Data Views Setup

To enable efficient exploration and visualization of Suricata and ModSecurity logs, custom data views (formerly called index patterns) were created in Kibana. Data views define the set of Elasticsearch indices from which Kibana reads data and specify the time field for time-based analysis.

- For Suricata alerts, the data view named "Suricata Alerts" was created using the pattern logs-suricata.alerts-so*. This data view aggregates all alert indices that Security Onion streams into Elasticsearch, enabling focused monitoring of network intrusion alerts.



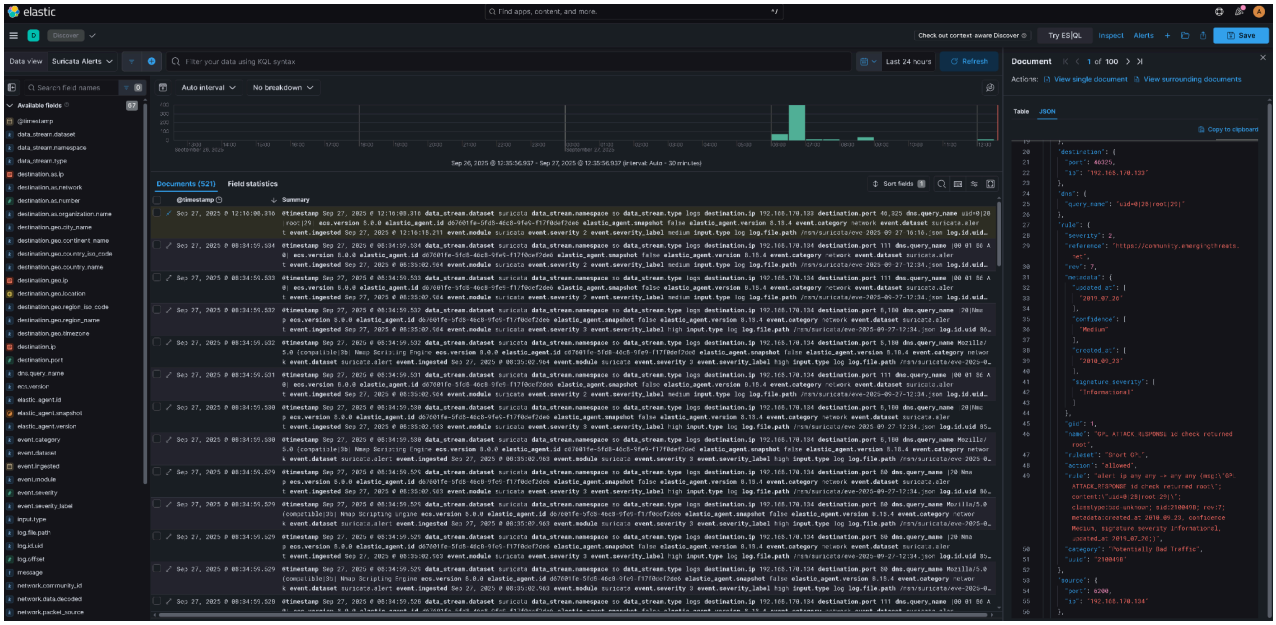

- For host and application logs forwarded via Filebeat (including ModSecurity logs), a data view named "Filebeat Logs" was created with the pattern logs-elastic_agent.filebeat-default*. This provides a consolidated view of host-based events and WAF activity.



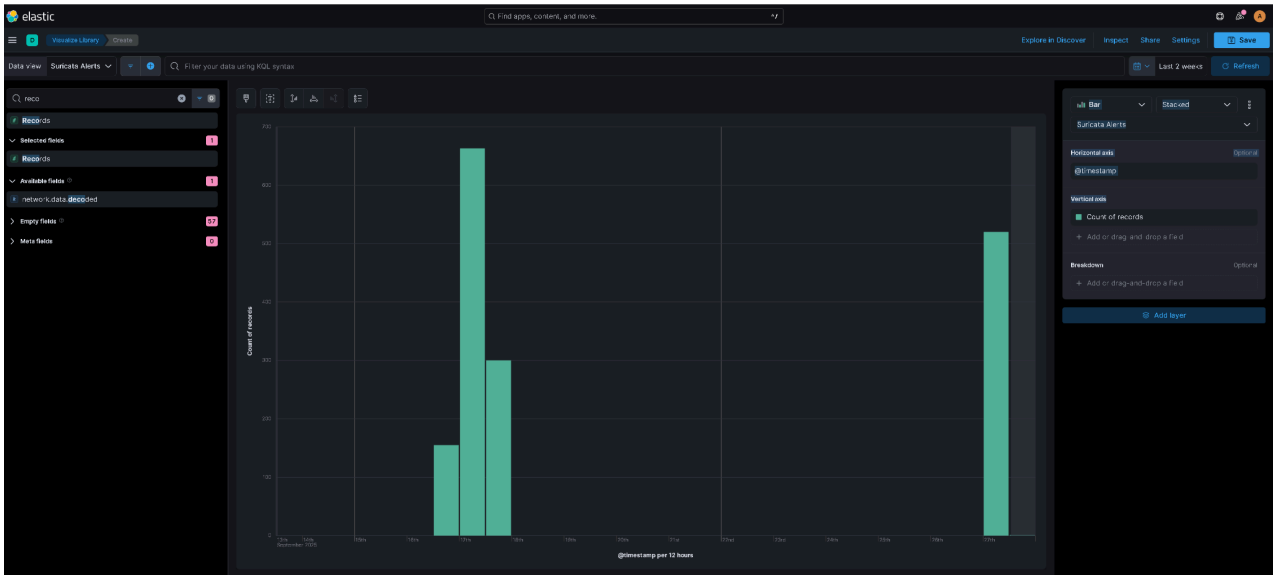

Both data views use the @timestamp field as the primary time filter, allowing temporal correlation and timeline visualizations of security events.

By isolating these datasets with tailored data views, Kibana enables detailed discovery queries, customized dashboards, and rapid incident investigation.

- Suricata and ModSecurity logs are sent to Kibana, where real-time alerts for brute-force and exploit attempts are displayed.



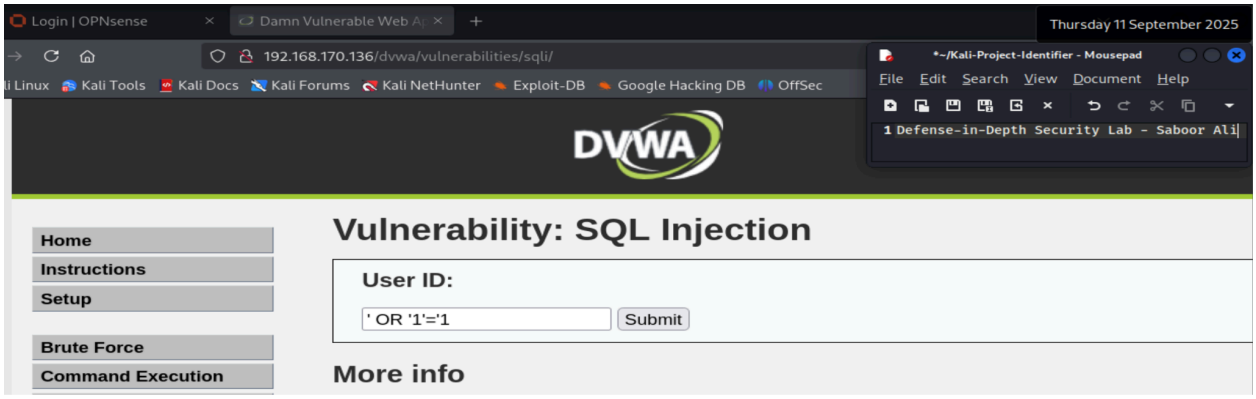
- FTP brute-force attempts, web attacks, and privilege escalation are all correlated in SIEM and visualized on a timeline.



## Post-Defense Attack Testing & Validation

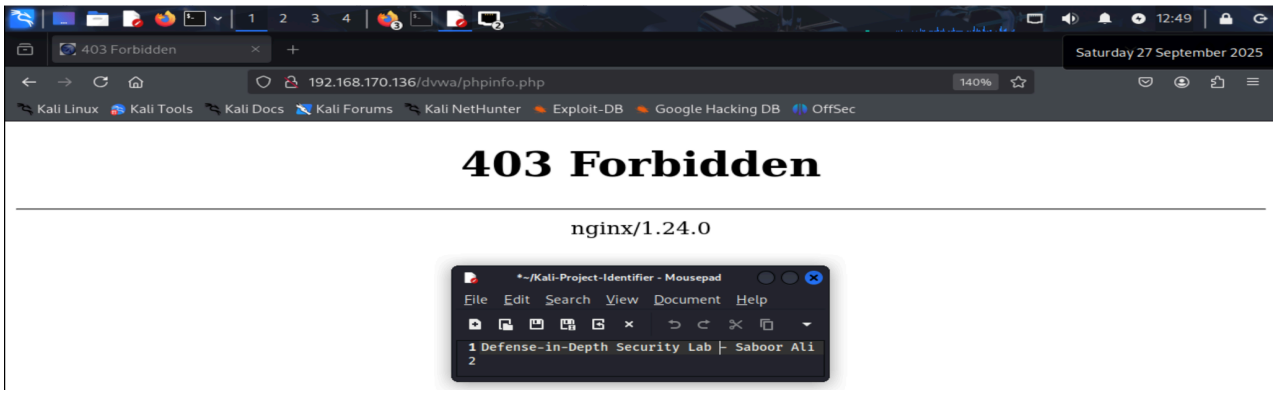
| Attack Type | Prevented? | Visible in SIEM? | Comments |
|---|---|---|---|
| SQL Injection | Yes | Yes | WAF blocks, alert logged |
| XSS | Yes | Yes | WAF blocks, alert logged |
| Directory Traversal | Yes | Yes | WAF blocks, alert logged |
| Nmap Scans | No | Yes | Detected, not blocked |
| FTP Brute Force/Backdoor | No | Yes | Alerts log shell attempts |

- Post-deployment testing verified that SQL Injection and Cross-Site Scripting attempts were intercepted and blocked by the WAF, returning HTTP 403 Forbidden responses.

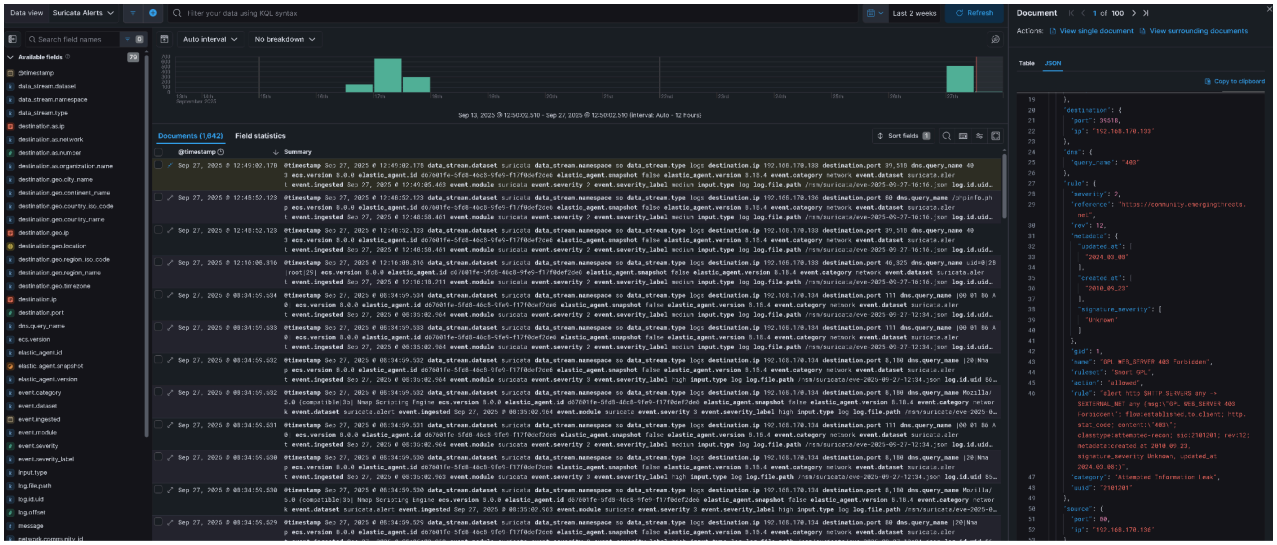




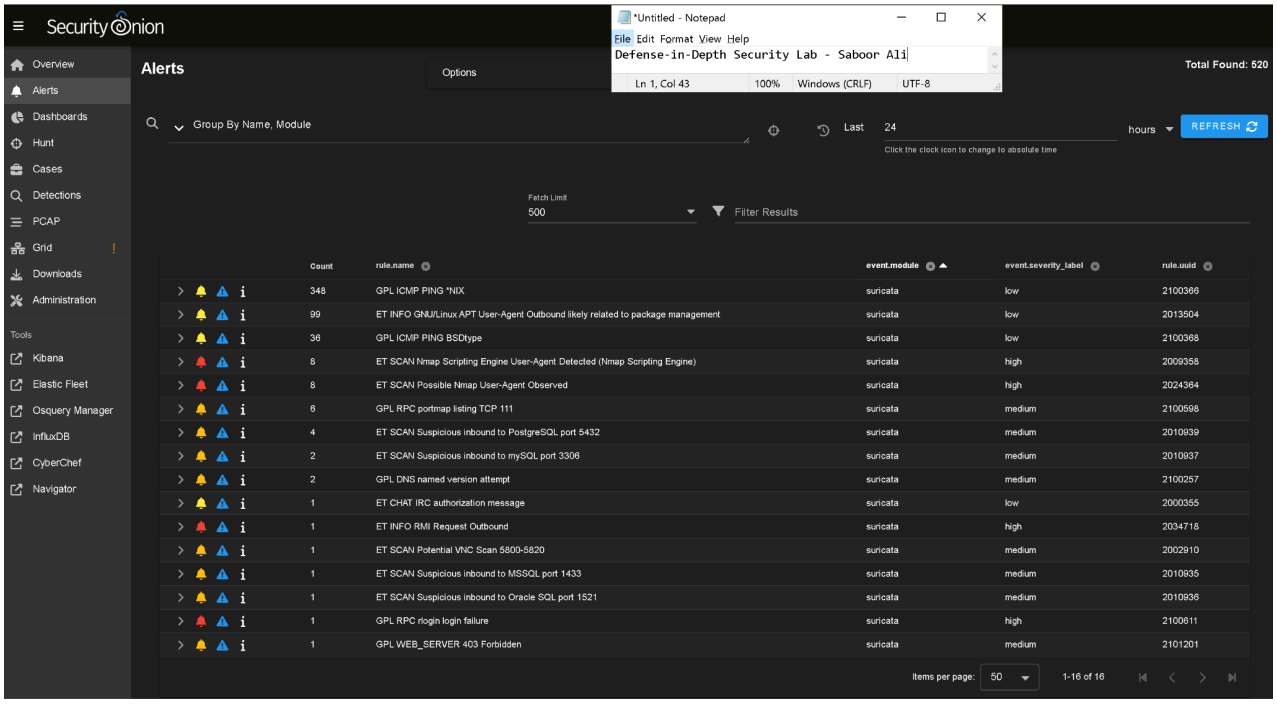
- **Directory Traversal**: ModSecurity logs and denies attack attempts for sensitive file and directory probes.



- Detected: Brute force and nmap attempts appear as alerts/events in SIEM, though prevention was incomplete.



- SIEM Forensics: Each attack, blocked or detected, is matched with corresponding forensic evidence in the SIEM dashboard.



## Limitations & Observations

- Certain attacks, such as Nmap scans and brute force attempt,s were effectively detected and correlated within the SIEM but not fully prevented by the existing controls.
- The defense-in-depth approach demonstrated is most effective against known web exploitation vectors (SQL Injection, XSS, Directory Traversal) while challenges remain for network reconnaissance and credential-based brute force attacks.
- This project represents the initial stage of security implementation. Further security automation and orchestration will be explored in upcoming phases using SOAR (Security Orchestration, Automation, and Response) platforms.
- Planned improvements include integrating SOAR capabilities for automated alert triage, coordinated incident responses, and potentially enhanced blocking mechanisms to cover gaps left by traditional prevention systems.

---

## Conclusion

This lab experimentally validates a robust layered security architecture combining network-level and application-layer controls with centralized security monitoring. Deploying OPNsense as a Next-Generation Firewall, ModSecurity as a Web Application Firewall, and Security Onion as a SIEM platform, this defense-in-depth model demonstrated the capacity to mitigate a broad spectrum of threats effectively.

Proactive configurations successfully blocked high-risk web attacks like SQL Injection, Cross-Site Scripting, and directory traversal attempts. At the same time, the SIEM environment enabled detection and correlation of reconnaissance and brute force activities, underscoring the importance of monitoring amidst evolving threat landscapes.

Visual tools such as the Kibana alert timeline allowed clear observation of attack patterns and temporal distributions, reinforcing data-driven analytics as central to modern cybersecurity practice. This layered approach reduces system exposure while enhancing visibility into attacker behaviors, creating a scalable and pragmatic security strategy.

Importantly, this project underscores the reality that no single control is sufficient; full resilience requires a fusion of prevention, detection, and continuous monitoring fortified by emerging technologies like SOAR. Future enhancements will leverage SOAR platforms to automate alert handling, orchestrate complex responses, and aim to mitigate currently unblocked attack vectors, further strengthening organizational defenses in an ever-shifting threat environment.