OWASP Web Attack Detection

Submission Deadline: 9 – jan - 2026
Submission Method: PDF

A. Title & Group Members

Project Title

OWASP Web Attack Detection

Intrusion Detection and Prevention System (IDPS) Using Suricata (OWASP-Based Project)

B. Objective of the Project

The objective of this project is to detect and analyze common OWASP web-based attacks using Suricata IDS. The project specifically focuses on testing and monitoring attacks such as Cross-Site Scripting (XSS), Local File Inclusion (LFI), Remote File Inclusion (RFI), Command Injection, and other malicious web traffic. By generating these attacks in a controlled lab environment, the project demonstrates how an IDPS identifies suspicious patterns and logs alerts for effective incident response.

C. Tools Used

Defensive / Monitoring Tools

- Ubuntu Linux (Suricata IDS Machine)
- Suricata IDS
- Emerging Threats Rule Set
- Custom `local.rules`

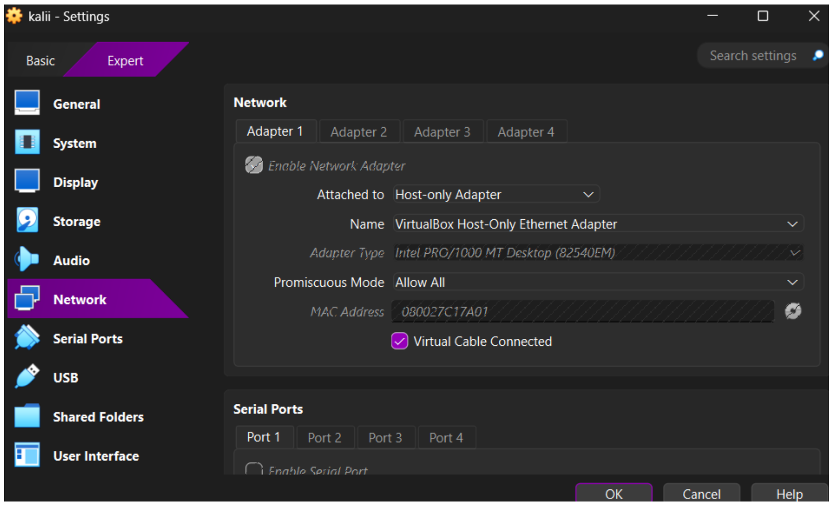
Attacking / Traffic Generation Tools

- Kali Linux
- Nmap (Port Scanning & Service Detection)
- Curl (HTTP Requests / Web Attacks)
- Metasploitable 2 (Vulnerable Target Machine)
- Nikto (For checking web vulnerability)

D. Step-by-Step Procedure

Step 1: Lab Environment Setup

- Create three virtual machines: Ubuntu, Kali Linux, and Metasploitable 2.
- Configure all machines in the same network.
- Assign Ubuntu as the monitoring machine.



kalii - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

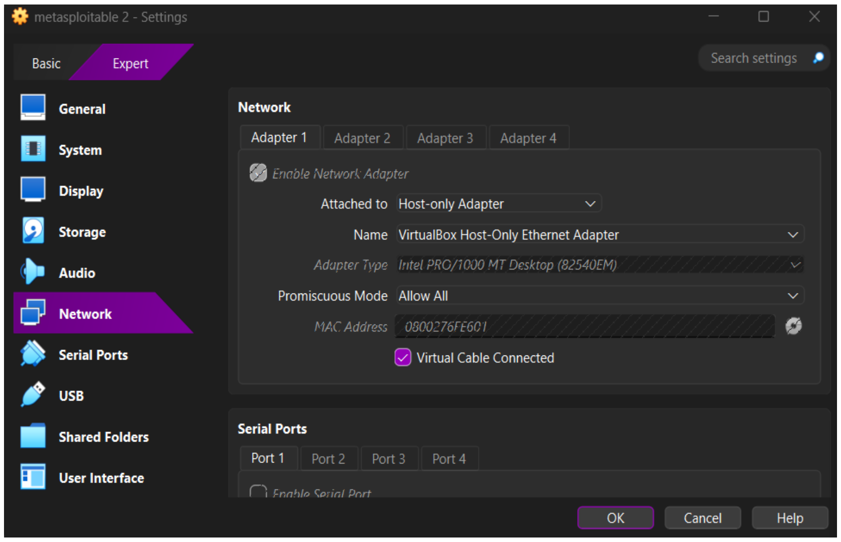
MAC Address: 080027C17A01

Virtual Cable Connected

Serial Ports

Port 1 | Port 2 | Port 3 | Port 4

Enable Serial Port



metasploitable 2 - Settings

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

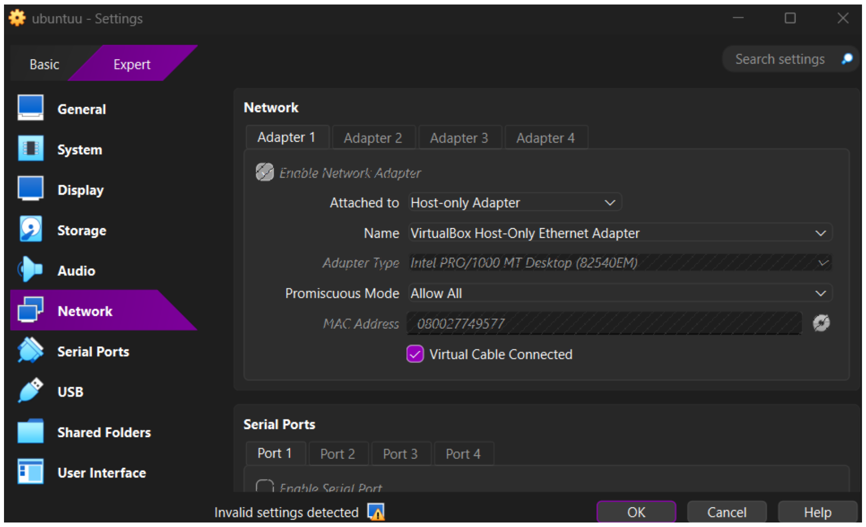
MAC Address: 0800276FE601

Virtual Cable Connected

Serial Ports

Port 1 | Port 2 | Port 3 | Port 4

Enable Serial Port



Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

MAC Address: 080027749577

☑ Virtual Cable Connected

Serial Ports

Port 1 | Port 2 | Port 3 | Port 4

Enable Serial Port

Invalid settings detected

Step 2: Install Suricata on Ubuntu

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:oisf/suricata-stable
sudo apt-get update

sudo apt install suricata
```

Step 3: Configure Suricata

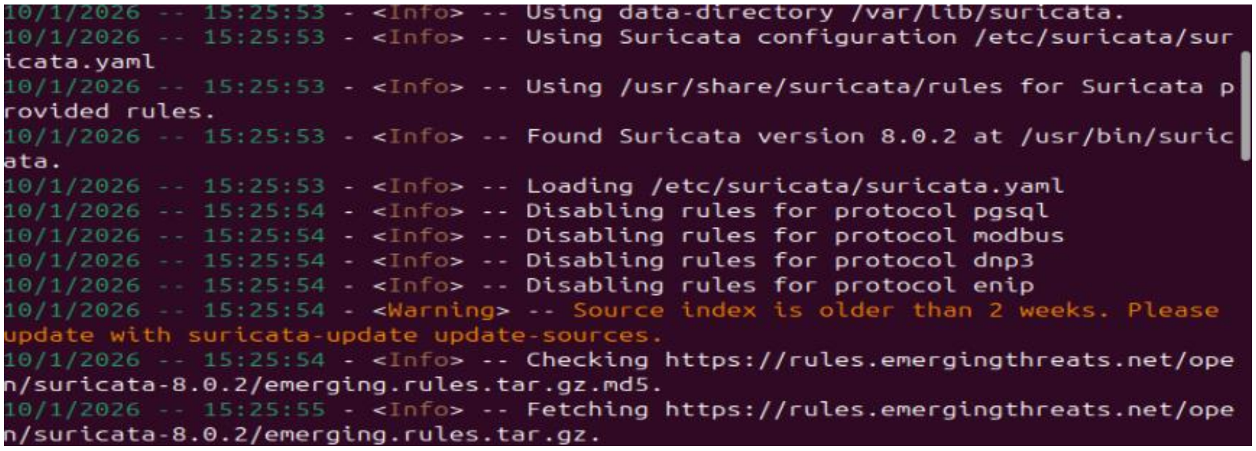
- Edit the Suricata configuration file:

```
sudo nano /etc/suricata/suricata.yaml
```

- Set correct HOME_NET value(192.168.213.3)
- Select the correct network interface. (enp0s3)

Step 4: Update Emerging Threats Rules

```
sudo suricata-update
```



```
10/1/2026 -- 15:25:53 - <Info> -- Using data-directory /var/lib/suricata.
10/1/2026 -- 15:25:53 - <Info> -- Using Suricata configuration /etc/suricata/sur
icata.yaml
10/1/2026 -- 15:25:53 - <Info> -- Using /usr/share/suricata/rules for Suricata p
rovided rules.
10/1/2026 -- 15:25:53 - <Info> -- Found Suricata version 8.0.2 at /usr/bin/suric
ata.
10/1/2026 -- 15:25:53 - <Info> -- Loading /etc/suricata/suricata.yaml
10/1/2026 -- 15:25:54 - <Info> -- Disabling rules for protocol pgsql
10/1/2026 -- 15:25:54 - <Info> -- Disabling rules for protocol modbus
10/1/2026 -- 15:25:54 - <Info> -- Disabling rules for protocol dnp3
10/1/2026 -- 15:25:54 - <Info> -- Disabling rules for protocol enip
10/1/2026 -- 15:25:54 - <Warning> -- Source index is older than 2 weeks. Please
update with suricata-update update-sources.
10/1/2026 -- 15:25:54 - <Info> -- Checking https://rules.emergingthreats.net/ope
n/suricata-8.0.2/emerging.rules.tar.gz.md5.
10/1/2026 -- 15:25:55 - <Info> -- Fetching https://rules.emergingthreats.net/ope
n/suricata-8.0.2/emerging.rules.tar.gz.
```

Step 5: Configure local.rules (Custom Rules)

Local rules are used to create custom detection rules based on specific attack behavior.

- Open local rules file:
- Add custom rules (e.g., Nmap scan detection, brute-force attempts).

alert icmp any any -> any any (msg:"ICMP Ping Detected"; sid:2000001; rev:1;)

alert tcp any any -> any any (flags:S; msg:"TCP SYN Scan Detected"; sid:2000002; rev:1;)

alert tcp any any -> any any (flags:F; msg:"TCP FIN Scan Detected"; sid:2000003; rev:1;)

alert tcp any any -> any any (flags:P; msg:"TCP PUSH Flag Detected"; sid:2000004; rev:1;)

alert tcp any any -> any any (msg:"Nmap Scan Detected"; content:"Nmap"; nocase; sid:2000010; rev:1;)

alert tcp any any -> any any (msg:"Masscan Detected"; content:"masscan"; nocase; sid:2000011; rev:1;)

alert udp any any -> any any (msg:"UDP Scan Detected"; sid:2000012; rev:1;)

alert tcp any any -> any 80 (msg:"HTTP Traffic Detected"; sid:2000020; rev:1;)

alert tcp any any -> any 80 (msg:"Nikto Scanner Detected"; content:"Nikto"; nocase; sid:2000021; rev:1;)

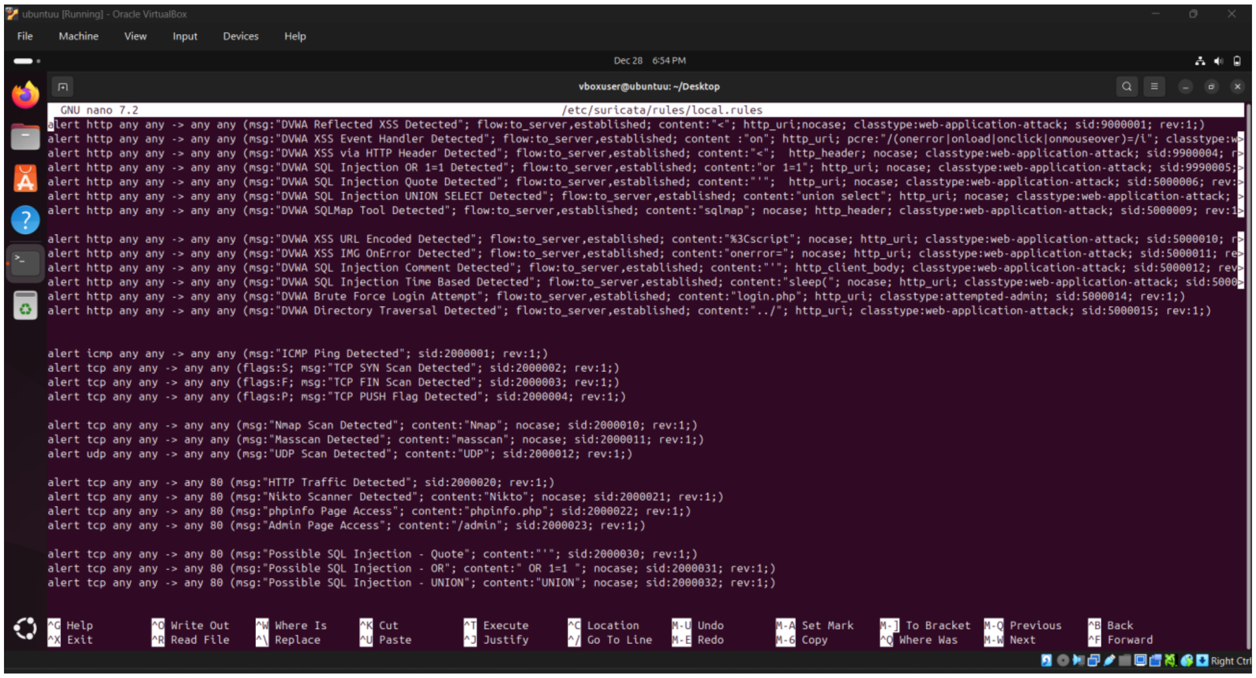
alert tcp any any -> any 80 (msg:"phpinfo Page Access"; content:"phpinfo.php"; sid:2000022; rev:1;)

alert tcp any any -> any 80 (msg:"Admin Page Access"; content:"/admin"; sid:2000023; rev:1;)

```
alert tcp any any -> any 80 (msg:"Possible SQL Injection - Quote"; content:"'"; sid:2000030; rev:1;)

alert tcp any any -> any 80 (msg:"Possible SQL Injection - OR"; content:" OR "; nocase; sid:2000031; rev:1;)

alert tcp any any -> any 80 (msg:"Possible SQL Injection - UNION"; content:"UNION"; nocase; sid:2000032; rev:1;)

alert tcp any any -> any 80 (msg:"Linux Command ls"; content:"ls"; sid:2000041; rev:1;)

alert tcp any any -> any 80 (msg:"Linux Command id"; content:"id"; sid:2000042; rev:1;)

alert tcp any any -> any any (msg:"Reverse Shell /bin/sh"; content:"/bin/sh"; nocase; sid:2000050; rev:1;)

alert tcp any any -> any any (msg:"Netcat Reverse Shell"; content:"nc -e"; nocase; sid:2000051; rev:1;)

alert tcp any any -> any any (msg:"Meterpreter Payload"; content:"meterpreter"; nocase; sid:2000052; rev:1;)

alert tcp any any -> any 21 (msg:"FTP Login Attempt"; sid:2000060; rev:1;)

alert tcp any any -> any 22 (msg:"SSH Connection Attempt"; sid:2000061; rev:1;)

alert tcp any any -> any 23 (msg:"Telnet Access Attempt"; sid:2000062; rev:1;)

alert http any any -> any any (msg:"DVWA Reflected XSS Detected"; flow:to_server,established; content:"<"; http_uri; nocase; classtype:web-application-attack; sid:7000001; rev:1;)

alert http any any -> any any (msg:"DVWA Stored XSS Detected"; flow:to_server,established; content:"<"; http_client_body; nocase; classtype:web-application-attack; sid:7000002; rev:1;)

alert http any any -> any any (msg:"DVWA XSS Event Handler Detected"; flow:to_server,established; content:"on"; http_uri; nocase; pcre:"/(onerror|onload|onclick|onmouseover)=/i"; classtype:web-application-attack; sid:7000003; rev:1;)

alert http any any -> any any (msg:"DVWA XSS via HTTP Header Detected"; flow:to_server,established; content:"<"; http_header; nocase; classtype:web-application-attack; sid:7000004; rev:1;)

alert http any any -> any any (msg:"DVWA SQL Injection OR 1=1 Detected"; flow:to_server,established; content:"or 1=1"; http_uri; nocase; classtype:web-application-attack; sid:7000005; rev:1;)

alert http any any -> any any (msg:"DVWA SQL Injection UNION SELECT Detected"; flow:to_server,established; content:"union"; http_uri; nocase; classtype:web-application-attack; sid:7000006; rev:1;)
```

alert http any any -> any any (msg:"DVWA SQL Injection Comment Detected"; flow:to_server,established; content:"--"; http_uri; classtype:web-application-attack; sid:7000007; rev:1;)

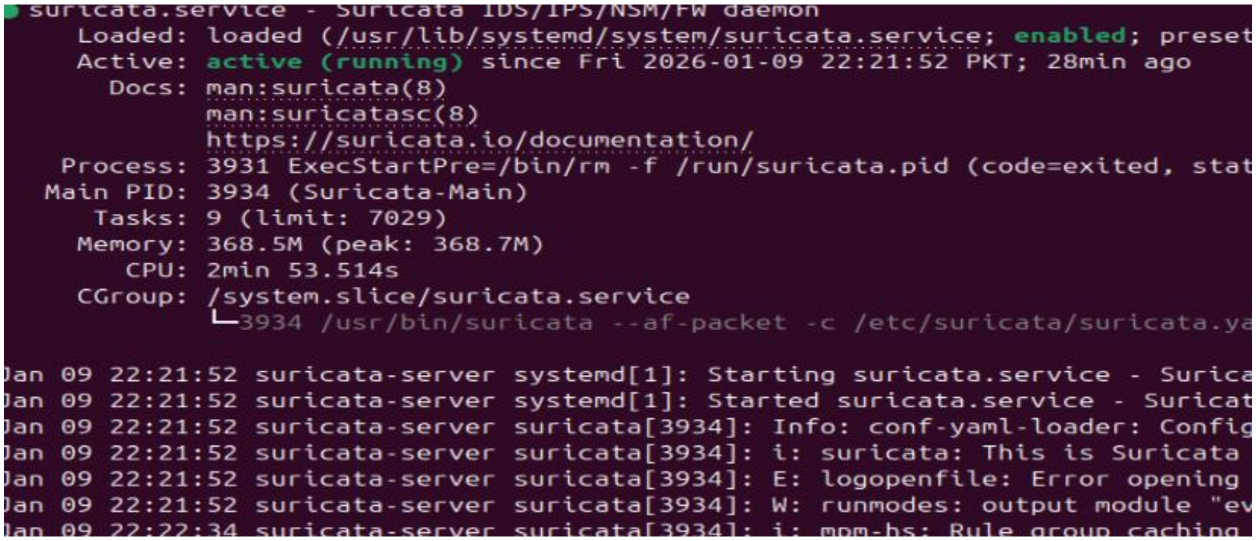
alert http any any -> any any (msg:"DVWA SQL Injection POST Body Detected"; flow:to_server,established; content:"'"; http_client_body; classtype:web-application-attack; sid:7000008; rev:1;)


ubuntuu [Running] - Oracle VirtualBox
File
Machine
View
Input
Devices
Help
Dec 28 6:54 PM
vboxuser@ubuntuu: ~/Desktop

```
  GNU nano 7.2                          /etc/suricata/rules/local.rules
alert http any any -> any any (msg:"DVWA Reflected XSS Detected"; flow:to_server,established; content:"<"; http_uri;nocase; classtype:web-application-attack; sid:9000001; rev:1;)
alert http any any -> any any (msg:"DVWA XSS Event Handler Detected"; flow:to_server,established; content :"on"; http_uri; pcre:"/(onerror|onload|onclick|onmouseover)=/i"; classtype:w>
alert http any any -> any any (msg:"DVWA XSS via HTTP Header Detected"; flow:to_server,established; content:"<";  http_header; nocase; classtype:web-application-attack; sid:9900004; r>
alert http any any -> any any (msg:"DVWA SQL Injection OR 1=1 Detected"; flow:to_server,established; content:"or 1=1"; http_uri; nocase; classtype:web-application-attack; sid:9990005;>
alert http any any -> any any (msg:"DVWA SQL Injection Quote Detected"; flow:to_server,established; content:"'";  http_uri; nocase; classtype:web-application-attack; sid:5000006; rev:>
alert http any any -> any any (msg:"DVWA SQL Injection UNION SELECT Detected"; flow:to_server,established; content:"union select"; http_uri; nocase; classtype:web-application-attack; >
alert http any any -> any any (msg:"DVWA SQLMap Tool Detected"; flow:to_server,established; content:"sqlmap"; nocase; http_header; classtype:web-application-attack; sid:5000009; rev:1>

alert http any any -> any any (msg:"DVWA XSS URL Encoded Detected"; flow:to_server,established; content:"%3Cscript"; nocase; http_uri; classtype:web-application-attack; sid:5000010; r>
alert http any any -> any any (msg:"DVWA XSS IMG OnError Detected"; flow:to_server,established; content:"onerror="; nocase; http_uri; classtype:web-application-attack; sid:5000011; re>
alert http any any -> any any (msg:"DVWA SQL Injection Comment Detected"; flow:to_server,established; content:"'"; http_client_body; classtype:web-application-attack; sid:5000012; rev>
alert http any any -> any any (msg:"DVWA SQL Injection Time Based Detected"; flow:to_server,established; content:"sleep("; nocase; http_uri; classtype:web-application-attack; sid:5000>
alert http any any -> any any (msg:"DVWA Brute Force Login Attempt"; flow:to_server,established; content:"login.php"; http_uri; classtype:attempted-admin; sid:5000014; rev:1;)
alert http any any -> any any (msg:"DVWA Directory Traversal Detected"; flow:to_server,established; content:"../"; http_uri; classtype:web-application-attack; sid:5000015; rev:1;)


alert icmp any any -> any any (msg:"ICMP Ping Detected"; sid:2000001; rev:1;)
alert tcp any any -> any any (flags:S; msg:"TCP SYN Scan Detected"; sid:2000002; rev:1;)
alert tcp any any -> any any (flags:F; msg:"TCP FIN Scan Detected"; sid:2000003; rev:1;)
alert tcp any any -> any any (flags:P; msg:"TCP PUSH Flag Detected"; sid:2000004; rev:1;)

alert tcp any any -> any any (msg:"Nmap Scan Detected"; content:"Nmap"; nocase; sid:2000010; rev:1;)
alert tcp any any -> any any (msg:"Masscan Detected"; content:"masscan"; nocase; sid:2000011; rev:1;)
alert udp any any -> any any (msg:"UDP Scan Detected"; content:"UDP"; sid:2000012; rev:1;)

alert tcp any any -> any 80 (msg:"HTTP Traffic Detected"; sid:2000020; rev:1;)
alert tcp any any -> any 80 (msg:"Nikto Scanner Detected"; content:"Nikto"; nocase; sid:2000021; rev:1;)
alert tcp any any -> any 80 (msg:"phpinfo Page Access"; content:"phpinfo.php"; sid:2000022; rev:1;)
alert tcp any any -> any 80 (msg:"Admin Page Access"; content:"/admin"; sid:2000023; rev:1;)

alert tcp any any -> any 80 (msg:"Possible SQL Injection - Quote"; content:"'"; sid:2000030; rev:1;)
alert tcp any any -> any 80 (msg:"Possible SQL Injection - OR"; content:" OR 1=1 "; nocase; sid:2000031; rev:1;)
alert tcp any any -> any 80 (msg:"Possible SQL Injection - UNION"; content:"UNION"; nocase; sid:2000032; rev:1;)

^G Help      ^O Write Out  ^W Where Is  ^K Cut       ^T Execute   ^C Location    M-U Undo  M-A Set Mark  M-] To Bracket  M-Q Previous  ^B Back
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line  M-E Redo  M-6 Copy      ^Q Where Was    M-W Next      ^F Forward
```

Step 6: Restart Suricata

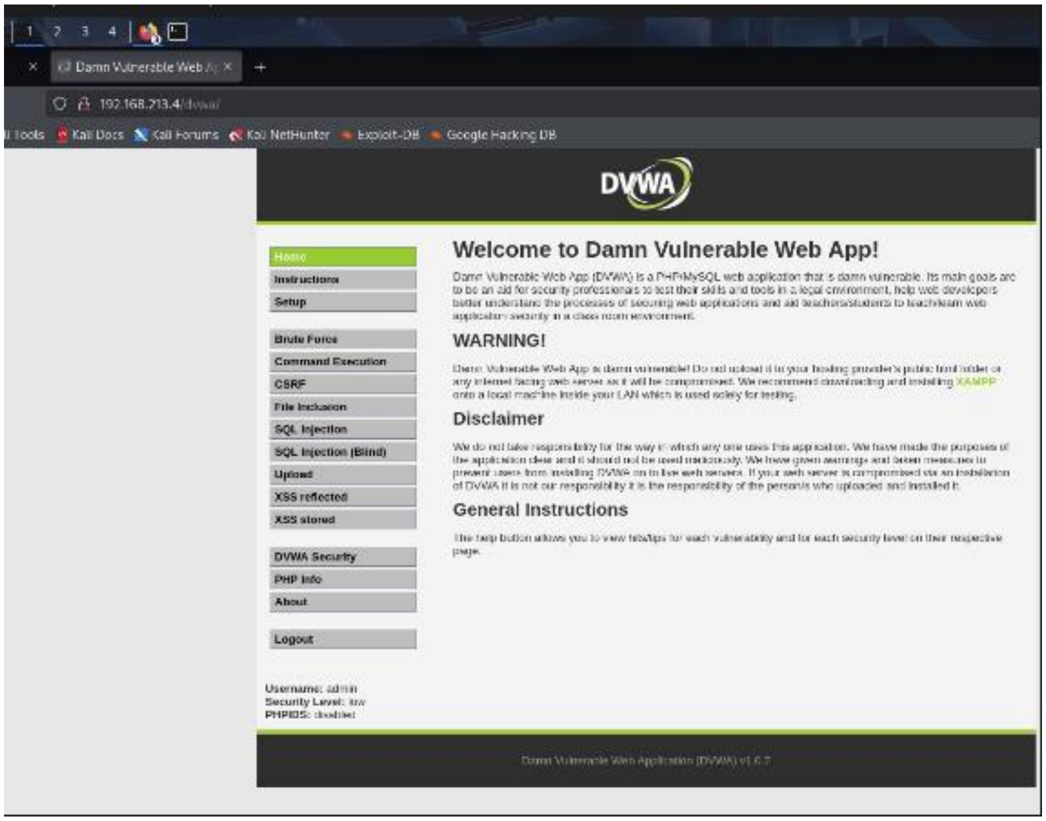
```
sudo systemctl restart suricata
```



```
● suricata.service - Suricata IDS/IPS/NSM/FW daemon
     Loaded: loaded (/usr/lib/systemd/system/suricata.service; enabled; preset
     Active: active (running) since Fri 2026-01-09 22:21:52 PKT; 28min ago
       Docs: man:suricata(8)
             man:suricatasc(8)
             https://suricata.io/documentation/
    Process: 3931 ExecStartPre=/bin/rm -f /run/suricata.pid (code=exited, stat
   Main PID: 3934 (Suricata-Main)
      Tasks: 9 (limit: 7029)
     Memory: 368.5M (peak: 368.7M)
        CPU: 2min 53.514s
     CGroup: /system.slice/suricata.service
             └─3934 /usr/bin/suricata --af-packet -c /etc/suricata/suricata.ya

Jan 09 22:21:52 suricata-server systemd[1]: Starting suricata.service - Surica
Jan 09 22:21:52 suricata-server systemd[1]: Started suricata.service - Suricat
Jan 09 22:21:52 suricata-server suricata[3934]: Info: conf-yaml-loader: Config
Jan 09 22:21:52 suricata-server suricata[3934]: i: suricata: This is Suricata
Jan 09 22:21:52 suricata-server suricata[3934]: E: logopenfile: Error opening
Jan 09 22:21:52 suricata-server suricata[3934]: W: runmodes: output module "ev
Jan 09 22:22:34 suricata-server suricata[3934]: i: mpm-hs: Rule group caching
```

Step 7: Generate Attack Traffic from Kali Linux

1. First we opened the DVWAPP on the kali machine and made its security level low as we can see it in the image



192.168.213.4/dvwa/

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

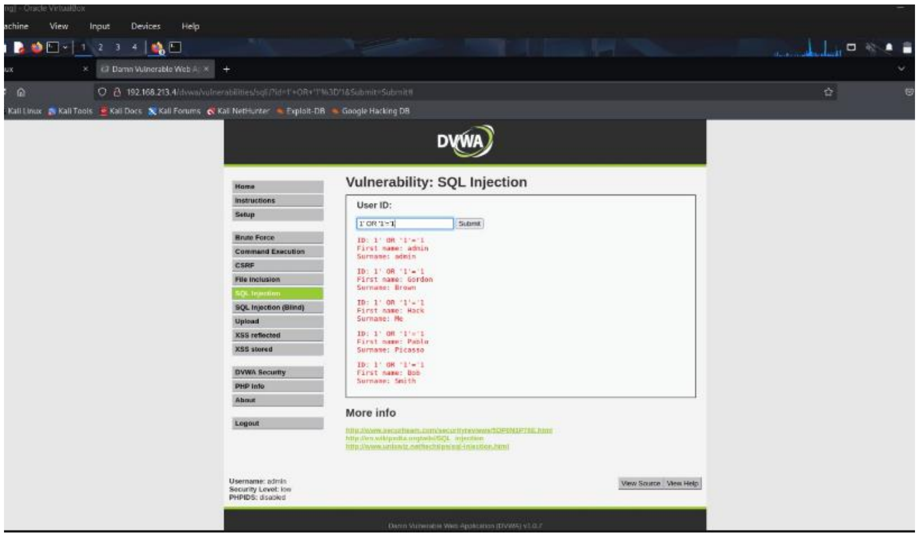
General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

2. Then we performed an sql injection which can be seen in the image below :



192.168.213.4/dvwa/vulnerabilities/sqli/?id=1'+OR+'1'%3D'1&Submit=Submit#

Vulnerability: SQL Injection

User ID: 1' OR '1'='1

```
ID: 1' OR '1'='1
First name: admin
Surname: admin

ID: 1' OR '1'='1
First name: Gordon
Surname: Brown

ID: 1' OR '1'='1
First name: Hack
Surname: Me

ID: 1' OR '1'='1
First name: Pablo
Surname: Picasso

ID: 1' OR '1'='1
First name: Bob
Surname: Smith
```

More info

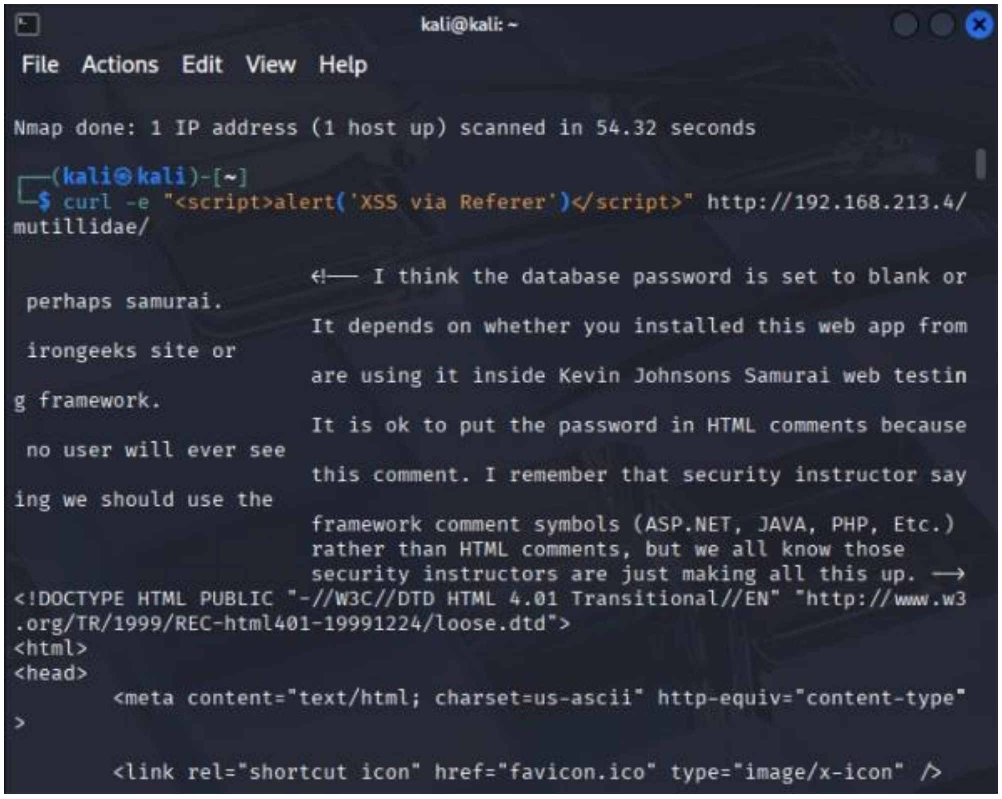
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

Log was generated against i which can be seen in the log section

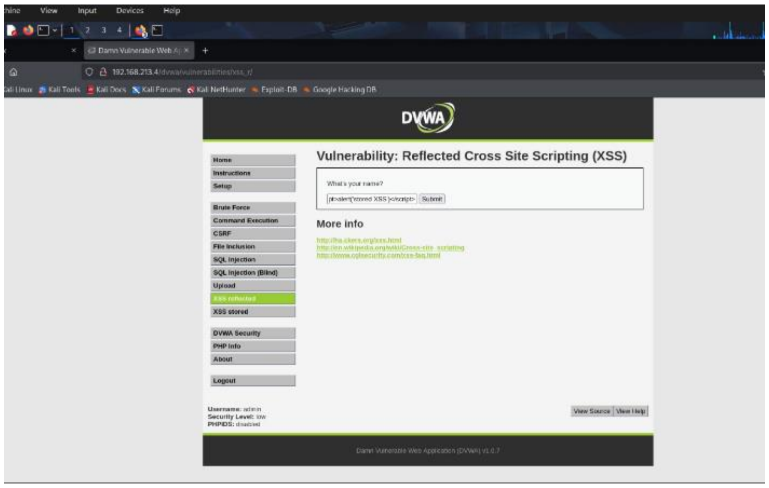
3. Then we performed another attack which is cross site scripting via http header method for this purpose we used the curl command



```
Nmap done: 1 IP address (1 host up) scanned in 54.32 seconds

┌──(kali㉿kali)-[~]
└─$ curl -e "<script>alert('XSS via Referer')</script>" http://192.168.213.4/
mutillidae/

                        <!—— I think the database password is set to blank or
 perhaps samurai.
                        It depends on whether you installed this web app from
 irongeeks site or
                        are using it inside Kevin Johnsons Samurai web testin
g framework.
                        It is ok to put the password in HTML comments because
 no user will ever see
                        this comment. I remember that security instructor say
ing we should use the
                        framework comment symbols (ASP.NET, JAVA, PHP, Etc.)
                        rather than HTML comments, but we all know those
                        security instructors are just making all this up. —>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3
.org/TR/1999/REC-html401-19991224/loose.dtd">
<html>
<head>
        <meta content="text/html; charset=us-ascii" http-equiv="content-type"
>

        <link rel="shortcut icon" href="favicon.ico" type="image/x-icon" />
```

Log was generated against it can be seen in the logs section

4. Now we performed xss again but this time from direct the web we inserted the payload in the input section on the web

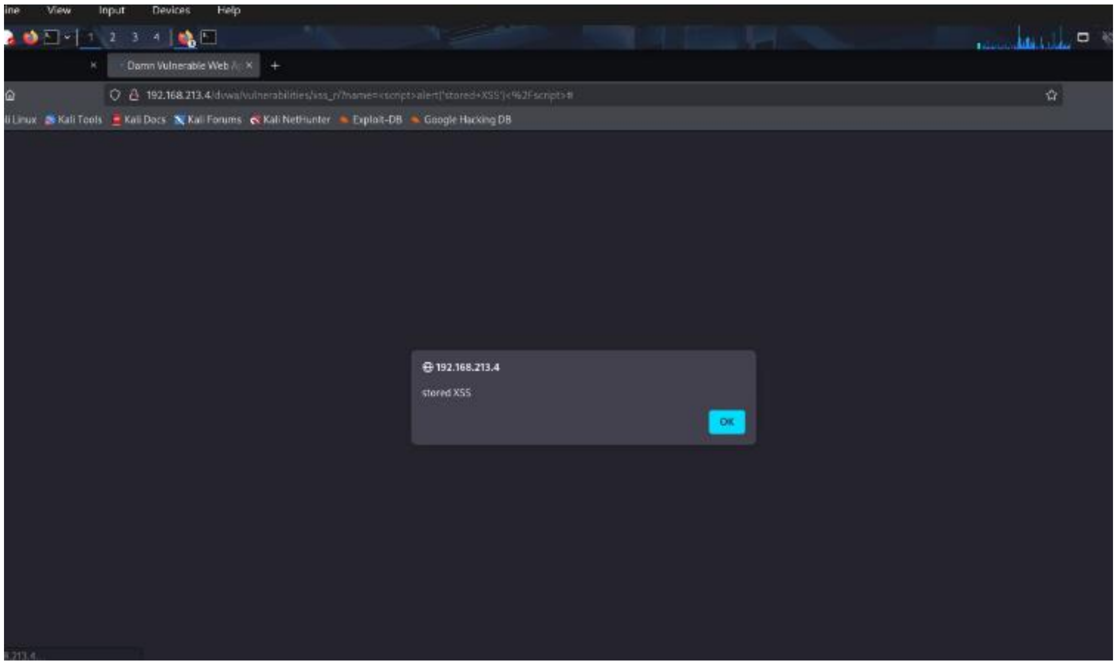


Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

pt>alert('stored XSS')</script>

More info



192.168.213.4/dvwa/vulnerabilities/xss_r/?name=<script>alert('stored+XSS')<%2Fscript>#

192.168.213.4

stored XSS

We can see that the payload was executed successfully

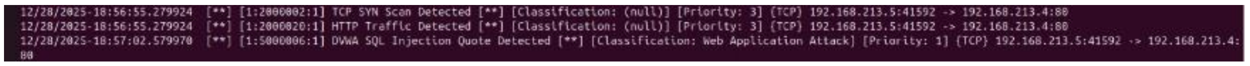
E. Logs & Evidence

fast.log

- Shows quick alert summaries generated by Suricata.

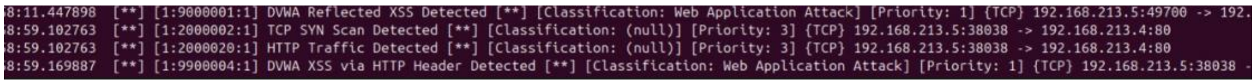
For this we used the command :

sudo tail -f /var/log/suricata/fast.log



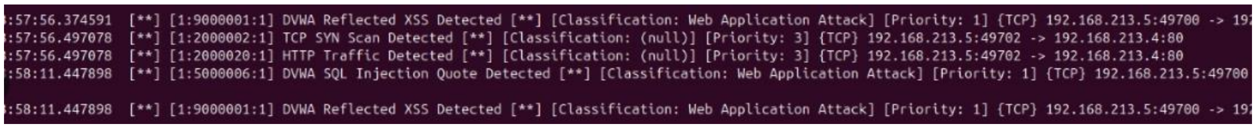
```
12/28/2025-18:56:55.279924  [**] [1:2000002:1] TCP SYN Scan Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.213.5:41592 -> 192.168.213.4:80
12/28/2025-18:56:55.279924  [**] [1:2000020:1] HTTP Traffic Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.213.5:41592 -> 192.168.213.4:80
12/28/2025-18:57:02.579970  [**] [1:5000006:1] DVWA SQL Injection Quote Detected [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.213.5:41592 -> 192.168.213.4:
80
```

The above log message shows that the rule was triggered against the SQL injection



```
8:11.447898  [**] [1:9000001:1] DVWA Reflected XSS Detected [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.213.5:49700 -> 192.
8:59.102763  [**] [1:2000002:1] TCP SYN Scan Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.213.5:38038 -> 192.168.213.4:80
8:59.102763  [**] [1:2000020:1] HTTP Traffic Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.213.5:38038 -> 192.168.213.4:80
8:59.169887  [**] [1:9900004:1] DVWA XSS via HTTP Header Detected [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.213.5:38038 -
```

In the above log image we can see that log was triggered for xss via http header



```
:57:56.374591  [**] [1:9000001:1] DVWA Reflected XSS Detected [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.213.5:49700 -> 19
:57:56.497078  [**] [1:2000002:1] TCP SYN Scan Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.213.5:49702 -> 192.168.213.4:80
:57:56.497078  [**] [1:2000020:1] HTTP Traffic Detected [**] [Classification: (null)] [Priority: 3] {TCP} 192.168.213.5:49702 -> 192.168.213.4:80
:58:11.447898  [**] [1:5000006:1] DVWA SQL Injection Quote Detected [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.213.5:49700

:58:11.447898  [**] [1:9000001:1] DVWA Reflected XSS Detected [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.213.5:49700 -> 19
```

In the above we can see the log that shows that the attacker tried to perform cross site scripting

---

F. Findings

- Multiple alerts were triggered during attack execution.
- Alerts were generated due to suspicious traffic patterns matching IDS rules.
- Some alerts were informational and not actual attacks (false positives).
- No major false negatives were observed during testing.

G. Conclusion

This project provided practical experience in detecting common OWASP web attacks using an Intrusion Detection and Prevention System based on Suricata. By simulating real-world attacks such as XSS, LFI, RFI, and Command Injection in a controlled lab environment, we gained a clear understanding of how malicious web traffic behaves and how IDS rules identify these patterns. The use of both default Emerging Threats rules and custom local.rules helped strengthen our detection capabilities.

Through this project, we learned the importance of proper IDS configuration, including correct network setup, HOME_NET definition, and interface selection. Analyzing logs such as fast.log and eve.json allowed us to understand alert details like SID, signature name, and severity. We also observed how false positives can occur and why careful rule tuning is necessary to improve accuracy.

Overall, this project enhanced our understanding of web application security and the OWASP Top 10 risks. It demonstrated how IDPS solutions play a vital role in early attack detection and incident response. The knowledge gained from this project is valuable for real-world cybersecurity environments, where continuous monitoring and proactive defense are essential for protecting web applications and networks.